



Designed for

Microsoft®
Windows® 95

InstallShield

express

PROFESSIONAL

USER'S
GUIDE



InstallShield Express Professional

User's Guide

<http://www.installshield.com>

Copyright Information

InstallShield Express Professional User's Guide

© 1996 InstallShield Corporation. All rights reserved.

This document is for information purposes only. InstallShield Corporation makes no warranties, expressed or implied. InstallShield and DemoShield are registered trademarks of InstallShield Corporation. InstallShield Corporation is a business unit of Stirling Technologies, Inc.

Other brand and product names are trademarks or registered trademarks of the respective holders. Microsoft is a registered trademark and Windows, Windows NT, the Windows logo, and the Windows NT logo are registered trademarks of the Microsoft Corporation.

Information in this document is subject to change without notice and does not represent a commitment on the part of InstallShield Corporation. The software described in this document is furnished under a license agreement. The software may be used only in accordance with the terms of this agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or retransmitted in any form or by any means, electronically or mechanically, including photocopying, recording, or information recording and retrieval systems, for any other purpose other than the purchaser's personal use, without the express written permission of InstallShield Corporation.

The InstallShield Express Professional User's Guide was written by Michael Byrne and Tim Woomert.
Initial Publishing 15 May 1996.
Printed in the USA.
Part No. 231-10001 4/96

Visit our website at <http://www.installshield.com>

Contents

Copyright Information	ii
Contents	iii
InstallShield Express Product Support	
Register Your Software	vii
Necessary Information for Technical Support	vii
The Best Ways To Contact Technical Support	viii
Contacting InstallShield.....	x
Chapter 1 Introduction	
InstallShield Corporation: The Premier Name in Windows Installation Technology.....	1-2
InstallShield3 -- The Worldwide Standard for Enabling Software Distribution to All Windows Platforms	1-3
InstallShield Express Custom Editions	1-6
InstallShield Express Professional – A Visual Difference	1-7
Installation	1-10
Document Conventions.....	1-12
Chapter Descriptions.....	1-12
InstallShield Express Documentation.....	1-14
User Interface.....	1-17
Ideas for Improvements	1-20
Chapter 2 A Quick Tutorial	
Getting in the Express Lane.....	2-2
Exercise 1 Creating an Installation in Under Five Minutes.....	2-3
Exercise 2 Creating a Unique Look for the Installation	2-11
Exercise 3 Adding Multiple Setup Types to the Installation	2-19

Exercise 4 Modifying Your Customer's Registry and Adding Customized Icons.....	2-28
--------------------------------------------------------------------------------	------

Chapter 3 Building Your Installation

Opening and Saving Your Setup Project File.....	3-2
What an Installation Does	3-8
How Express Helps You Create Your Installation.....	3-8
Groups, Components, and Setup Types.....	3-9
Target Directory vs. Source Directory	3-19
Express Directory Specifiers	3-20
Creating a Windows 95 Compatible Application.....	3-24
Creating a Single File Executable	3-27
Version Control	3-29

Chapter 4 InstallShield Objects

What is an InstallShield Object?	4-2
Borland C++ 5.0 Objects.....	4-4
Delphi 2.0 Objects.....	4-7
Paradox 7.0 Objects.....	4-12
Visual Basic 4.0 Objects	4-15
Visual C++ 4.1 Objects	4-19
Advanced Settings	4-21
BDE/IDAPI Settings	4-22
SQL Links Settings	4-27
Selecting Individual ODBC Drivers	4-28

Chapter 5 InstallShield Extensions

Introduction	5-2
Adding an Extension	5-2

Calling a Custom .DLL	5-4
Running an .EXE File.....	5-5
Saving Space on Disk 1 for Uncompressed Files	5-7

Chapter 6 The Registry

What is the Registry?	6-2
Keys and Values	6-5
Your Installation and the Registry	6-8
Creating Custom Keys and Values	6-10
Removing Custom Registry Keys	6-14
Registry Editor.....	6-15

Chapter 7 Frequently Asked Questions

BDE.....	7-2
Create Distribution Media.....	7-3
Custom Installations.....	7-3
Disk Builder	7-4
Folders and Icons.....	7-6
ODBC.....	7-6
The Registry.....	7-6
Running the Installation.....	7-7
Self-registration of .DLLs and .OCXs	7-9
Setup Groups and Components.....	7-10
Uninstallation.....	7-12

Chapter 8 Express Dialog Boxes

The New Project Dialog Box	8-2
Set the Visual Design.....	8-4
Select InstallShield Objects.....	8-9

Specify Components and Files 8-18

Select User Interface Components 8-23

Make System File Changes 8-45

Make Registry Changes 8-50

Specify Folders and Icons 8-54

Run Disk Builder 8-58

Test the Installation 8-61

Create Distribution Media 8-61

Appendix A Glossary A-1

Appendix B InstallShield Express License Agreement B-1

Appendix C Disk Builder Warning Messages

Warning Messages C-1

Error Messages C-7

Appendix D Index D-1

InstallShield Express Product Support

InstallShield Corporation is committed to providing the best possible technical support for Express and all of our products. In order to get you the fastest, most complete, and most accurate support, please follow the procedures listed below.

Register Your Software

Completion of the InstallShield Express product registration form ensures that you are eligible for the benefits available to registered users, such as special offers, updates, and exclusive InstallShield Express information. By registering, you can also access our superior technical support staff free for thirty days to help you with any Express questions you might encounter. Express enables you to register by either of the following methods:

- If you elected not to register your copy of Express when you were installing the product, you can still register electronically. Select **Register** from the Help menu to launch the Online Registration process.
- If you have not already sent the registration form included with Express, please fill out and fax it to InstallShield Corporation at (847) 240-9120.

Remember that registering your copy of Express also lets us know where to send you maintenance releases and other important information as it becomes available.

Necessary Information for Technical Support

The technical support department at InstallShield likes to provide detailed and accurate answers to all of your questions. In order to do so, our representatives will need the following information:

- Your full name and the name of the registered user, if different.
- Your street address, phone and fax number, and email address.
- The serial number of your copy of Express.

- The full, six-digit Express version number. This can be found in the About box, which you can quickly open by clicking the **About** push button.
- The system information of the computer you are using, including the make and model of the machine and video driver, the operating system, the amount of memory and system resources, any relevant devices or peripherals, and any system .DLLs you are using.
- A detailed description of the problem. Describe any error messages exactly as they appear. Please list also the steps and conditions which led to the problem.

For your convenience, we have included a Technical Support Request Form which you can use to quickly organize the information we need to give you the quickest and best technical support. To access it, select **Technical Support** from the Help menu. You can use a non-standard form, just make sure that you include all of the necessary information listed earlier in this section.

The Best Ways To Contact Technical Support

Because of the amount of information our technical support department needs, and because finding an answer may require some research on our part, calling us is often the least effective method of getting technical support. We recommend the following methods as the quickest and best ways to get the answers you need:

1. **Express Documentation**
Before you spend the time gathering information, contacting InstallShield, and waiting for a response, make sure that the answer to your question is not in Express documentation. Take advantage of the search features in our help files. If you click the **Search** push button in any of Express's Help windows and go to the **Find** tab, you can create a full text help index and search for individual words or phrases.

Check the Readme file for any last-minute information and FAQs.

2. **Internet**
To keep informed of all the late breaking information relating to Express, consult the InstallShield Express website at <http://www.installshield.com/express>. This area is constantly being

updated with news and notes about Express. If you have a question about Express, you might save yourself time by checking this website for the answer.

If you are running Express on a Windows 95 system, you can launch your web browser and connect automatically with the Express website from within Express by either selecting

[**http://www.installshield.com/express**](http://www.installshield.com/express) from the Help menu or clicking the hypertext link located in the lower right corner of the main Express window.

3. Email

Once you have determined that both the Express documentation and the website do not contain the answer you need, the best way to contact the technical support department is through email. Our address is express@installshield.com.

4. Fax

Faxing is the next most efficient means of contacting us. Our technical support fax number is (847) 240-9138.

Contacting InstallShield

This section contains information that you might find helpful if you need to contact InstallShield for any reason.

InstallShield Express Support via Internet or Email

Description	Phone, Address, or ID
InstallShield Express WWW site	http://www.installshield.com/express
Technical support via email	express@installshield.com
Suggestions for future improvements in Express	isxwish@installshield.com

Technical Support

Description	Phone, Address, or ID
Monday through Friday 9:30 AM to 4:30 PM Central Time	Phone (847) 240-0041
Fax requests for technical support to this number.	FAX (847) 240-9138
Mail requests for technical support to this address.	Mailing Address Technical Support Department InstallShield Corp. 1100 Woodfield Drive Suite 108 Schaumburg, IL 60173-9946

Sales, Marketing, and Licensing

Description	Phone, Address, or ID
Call either number to order or request information about all InstallShield products.	Phone (800) 374-4353, (847) 240-9111
	FAX (847) 240-9120
	Email info@installshield.com
	Address

Chapter 1

Introduction

Congratulations on purchasing the Professional Edition of InstallShield Express. This chapter will introduce you to Express and explain the structure of this guide.

Contents

Welcome to InstallShield Express.....	1-2
Installation	1-10
Document Conventions.....	1-12
Chapter Descriptions	1-12
InstallShield Express Documentation	1-14
User Interface.....	1-17
Ideas for Improvements.....	1-20

InstallShield Corporation: The Premier Name in Windows Installation Technology

Based in Schaumburg, Illinois, InstallShield Corporation was founded in 1987 with the mission of developing applications and software development tools for emerging GUI operating systems.

InstallShield Corporation first introduced InstallShield version 1.0 in 1990 at the same time Microsoft introduced Windows 3.0. InstallShield 1.0 was the first graphical, native-Windows installation development system.

Within a year, InstallShield had gained recognition as a leader in installation technology, specializing in providing solutions for the growing Windows marketplace. 1991 saw the release of Windows 3.1 and InstallShield 2.0, which contained significant improvements and enhancements over version 1.0.

InstallShield 2.0 established itself as the worldwide standard for Windows installations, used to distribute over 20,000 commercial applications.

In 1994, Microsoft chose to work exclusively with InstallShield Corporation to develop an installation system for their upcoming Windows 95 operating environment. Microsoft and InstallShield worked together to define the requirements and procedures for creating Windows 95 setup programs. As a result of this relationship, Microsoft licensed the InstallShield SDK Edition to be included in the Windows 95 Software Developer's Kit and Microsoft Visual C++ 4.0.

InstallShield Corporation released InstallShield3 in July 1995, and it quickly became the installation system of choice for most major commercial and corporate applications designed for Windows 95.

InstallShield3 -- The Worldwide Standard for Enabling Software Distribution to All Windows Platforms

Two versions of InstallShield3 Professional Edition are available: InstallShield3 for Windows 3.1 (16-bit InstallShield), and InstallShield3 for Windows 95 and Windows NT (32-bit InstallShield, available for Intel, MIPS, Alpha, or Power PC).

Based on your application's installation requirements, you can use the 16-bit or the 32-bit product, or the 16-bit and 32-bit products can be combined on a single floppy disk or CD-ROM for cross-platform installations.

InstallShield3 for Windows 3.1

InstallShield3 for Windows 3.1 is a 16-bit Windows 3.1 application that runs under all versions of Windows. InstallShield3 for Windows 3.1 can be combined with InstallShield3 for Windows 95 and Windows NT to offer seamless installation system for mixed Windows (cross-platform) installations. If you are only targeting Windows 3.1, then you need InstallShield3 for Windows 3.1.

InstallShield3 for Windows 95 and Windows NT

InstallShield3 for Windows 95 and Windows NT is a native 32-bit Windows application designed to install 32-bit Windows applications tightly integrated with the 32-bit operating system. If you are targeting Windows 95 or Windows NT, then you need InstallShield3 for Windows 95 and Windows NT.

Both versions of InstallShield can be combined on a floppy disk or CD-ROM to provide a cross-platform solution for your applications. You do not need to write two different scripts. A single script can be used to install your application correctly onto any platform.

InstallShield3 gives you the power to create bulletproof, comprehensive, customized installation and uninstallation systems for your applications. And InstallShield3 now includes an Unlimited Application License, allowing you to write installations for any number of Windows applications!

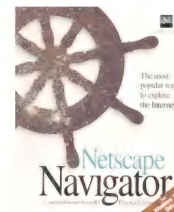
Here are just some of the features:

- The ability to create a single installation system that will tightly integrate with Windows 3.1, Windows for Workgroups, Windows 95, and Windows NT 3.5.
- Comprehensive installation templates.
- InstallShield Wizard EU with more than 50 built-in user interface objects, and complete customization of all objects.
- The ability to integrate your own custom dialog boxes.
- InstallShield Visual Debugger, a Windows hosted debugger specifically created for installation building and testing.
- Use of your company logo, corporate look, animated bitmaps, and billboards during installation.
- Selectively install components of a multi-component application customized for each user.
- Launch help files, README files, other applications at any time during or after the installation.
- Intelligently update or modify system files, AUTOEXEC.BAT and CONFIG.SYS, private and system INI files and the Registry.
- Total extensibility to call any function, in any DLL, including any Windows API directly from the script.

The Clear Choice of World-class Developers

InstallShield Corporation has a greater than 85% market share in the Windows software installation category. The list of world-class companies and applications that have used InstallShield3 comprises a veritable who's-who of Windows software developers:

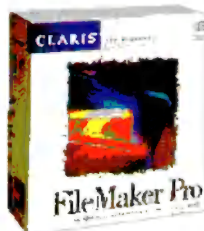
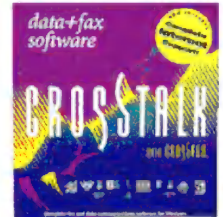
Netscape Communications Corporation uses InstallShield to deliver Netscape Navigator™.





CompuServe Incorporated picked InstallShield to create installations for its WinCIM™ and Navigator software.

AttachMate Corporation selected InstallShield as the installation system for CrossTalk™ and EXTRA! Personal Client 6.0™.



Claris Corporation chose InstallShield3 to distribute FileMaker Pro™.

Macromedia Inc. uses InstallShield on products such as ACTION! For Windows™ and Director 4.0.



Micrografx Corporation created the installation for ABC Graphic Suite™ with InstallShield.

Berkeley Systems distributes its After Dark™ screen saver using an InstallShield setup.



For a comprehensive list of leading InstallShield users, visit our Customer Gallery at <http://www.installshield.com>.

InstallShield International

InstallShield3 has a resource architecture and internationalization process that makes it easy for software developers to create installation programs in a variety of languages (see list below). There are two versions of InstallShield International: InstallShield International East and InstallShield International West. Both versions of InstallShield International include built-in resources, including the Sd dialogs, localized into each of the languages listed, and all essential InstallShield3 features and capabilities.

Using InstallShield International, you can produce installations in 28 languages:

East

Arabic, Chinese (Simplified), Chinese (Traditional), Hebrew, Indonesian, Japanese, Korean, Russian, Thai, and Turkish

West

Basque, Catalan, Czech, Danish, Dutch, Finnish, French, German, Greek, Hungarian, Italian, Norwegian, Polish, Portuguese (Brazilian), Portuguese (Standard), Slovenian, Spanish, and Swedish.

For more information regarding InstallShield International, visit our website at <http://www.installshield.com>.

InstallShield Express Custom Editions

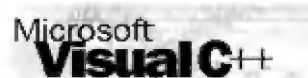
The InstallShield Express Business Unit of InstallShield Corporation was formed in 1994 with the goal of creating a powerful installation system that is ridiculously easy to use. The first step was the creation of InstallShield Express custom editions, a series of 32-bit visual installation development systems, each optimized for a particular development environment.



InstallShield Corporation furthered its strategic relationships with the development of InstallShield Express Delphi 2.0 Edition, the first of many collaborations between Borland and InstallShield. Borland and InstallShield have worked together closely to provide the most sophisticated technology available for properly installing Borland products and technologies.

Since then, InstallShield Express custom editions have been included with products such as Borland C++ 5.0 and Paradox 7 Client/Server, both domestic and international editions. Many more joint efforts are planned for deploying Visual dBASE, Local InterBase, and other Borland products.

InstallShield Corporation has also developed custom editions for Microsoft Visual Basic 4.0 and Microsoft Visual C++ 4.x. The Microsoft Visual C++ edition was included with the Visual C++ 4.1 and 4.2 CD's.



More custom editions are planned for the future as the Express team continues to provide more complete solutions for the installation needs of Windows developers. The additional functionality created for each custom edition is incorporated into InstallShield Express Professional

InstallShield Express Professional – A Visual Difference

InstallShield Corporation is proud to introduce InstallShield Express Professional.

Express Professional is a completely visual installation development system. It is designed to complement InstallShield3, providing a quick method for producing setups for your applications.

Express Professional combines the 32-bit installation-smart functionality of the custom editions with 16-bit support and dozens of valuable features to give you added power and flexibility.

Add Components Easily With InstallShield Objects

InstallShield Objects are one of the key component in Express. Each object encapsulates the necessary knowledge to deploy third-party components, such as Data Access Objects (DAO) and the Borland Database Engine (BDE). Each Express custom edition includes InstallShield Objects which allow you to redistribute vital dependent technology from the specified development environment by simply selecting a check box.

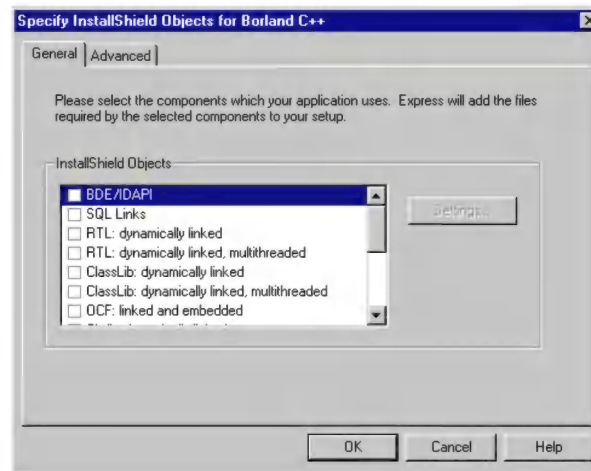


Figure 1.1. An InstallShield Objects sample dialog box.

InstallShield Express Professional Puts It All Together

InstallShield Express Professional contains all of the InstallShield Objects included in each of the custom editions, as well as many more valuable features:

- Full install and uninstall support for Windows 95, Windows NT, and Windows 3.1.
- Customizable user interface, including custom bitmaps and billboards, to give your installation a distinctive touch.
- Unlimited, lifetime distribution license, allowing you to use Express Professional as many times as you like.

- CD-ROM or disk installation options let you specify the distribution medium. Express automatically compresses your application files and splits them across whatever size medium you desire.
- Single-file, self-extracting .EXE option allows for easy distribution across networks of all types.
- Password protection for your single-file installations saves you the time and trouble of creating your own encryption algorithm – just type your password and Express takes care of it for you.
- Totally automated file compression and compressed library splitting. All you do is click a button and watch Express go.
- Online Registration gives you the power of state-of-the-art customer data management.
- Automatic registration of all standard OLE controls. You don't even have to specify that your OCX's and DLL's are self-registering– Express determines it for you.
- Automatic registry entries for Windows 95, helping you to get Microsoft's Designed for Windows 95 logo.
- Quick registry key and registry value creation. There's even a dialog which allows you to easily merge a .REG file.
- Default settings which fully conform to Windows 95 Setup Guidelines.
- Custom and Compact setup type options give you the power to create flexible installations.
- Easy icon specification allows you to specify your executable, parameters, .ICO file, and working directory, all through a point-and-click interface.
- The extensibility to call an external .DLL function during your installation. Anything you can include in a DLL, you can include in your setup.
- The ability to launch an executable during your installation. You can use this feature to run a program in the background, or even launch another installation. Express includes an easy way to specify the text to prompt your user for a separate set of disks, if needed.

- Full support for 16-bit and 32-bit ODBC. Select and modify the drivers and data sources you want to install.
- Simple modification of private and system .INI files, as well as AUTOEXEC.BAT and CONFIG.SYS.

You get all this, plus the peace of mind that comes with using genuine InstallShield products, which have been used by most leading software vendors to create millions of bulletproof, professional-quality installations.

Because InstallShield is the worldwide industry standard for software distribution to Microsoft Windows 3.1, Windows for Workgroups, Windows 95, and Windows NT, most Windows customers are familiar and comfortable with the non-technical interface of an InstallShield-created installation.

When you use InstallShield Express Professional, your customer will immediately associate your product with other world-class software applications, increasing his or her confidence in your product.

We invite you to install and launch Express so you can see for yourself why we're excited to be bringing this world-class product to you.

Installation

System Requirements

Please check that the system on which you will be installing Express is running either of the following operating systems:

- Windows 95
- OR-
- Windows NT version 3.51 or higher

The minimum requirements for Express are the same as for either of the above operating systems.

In order to create your installation with Express, you will need to have enough available space to hold two additional copies of your application on the drive where you are building the setup.

Installing Express on Your System

The InstallShield Express Professional installation will take approximately five to ten minutes, depending on your system's resources.

The Express CD-ROM is AutoPlay-enabled. If your system includes the AutoPlay feature, just insert your InstallShield Express Professional CD-ROM and the setup will be launched automatically.

If your system does not have AutoPlay, follow these instructions:

Windows 95 or Windows NT 4.0

1. Insert the Express CD-ROM.
2. Open the Control Panel window and click on the Add/Remove Programs icon.
3. Follow the installation instructions that appear on the screen.

Note We strongly recommend that you install all programs in Windows 95 or Windows NT 4.0 through the Add/Remove Programs function.

Windows NT 3.51

1. Insert the Express CD-ROM.
2. In File Manager, select the drive containing the Express CD-ROM.
3. In the Disk1 directory, launch SETUP.EXE.

The Express icon will be added to your Start Programs menu (Windows 95 or NT 4.0) or the Express program group (Windows NT 3.51).

Online Registration

InstallShield Express includes online registration, allowing you to register your copy of Express automatically using your modem. The online registration will be launched near the end of the Express setup.

You can include this same online registration with your own installation. Contact Pipeline Communications, Inc. at 1-800-WIN95REG, or by email at WIN95REG@pcpipeline.com for more information.

Document Conventions

The following table lists the terms, signs, and styles used throughout the InstallShield Express Professional User’s Guide.

Convention	Description
Italics	Italics denote the first occurrence in the text of a term found in the glossary. Also used for chapter and book titles.
Bold	Denotes emphasized terms and the titles of dialog boxes, menus, and command buttons.
User-entered information	This monospace style indicates information that the user types into an entry field.
Windows	Windows means any one of the following platforms: Microsoft Windows 3.1, Microsoft Windows for Workgroups 3.11, Microsoft Windows NT, or Microsoft Windows 95.
Windows 3.1	Windows 3.1 specifically refers to version 3.1 of Microsoft Windows only.
Windows NT	Windows NT specifically refers to Microsoft Windows NT 3.51 or higher only.
Windows 95	Windows 95 specifically refers to Microsoft Windows 95 only.

Chapter Descriptions

The InstallShield Express Professional User’s Guide has been designed to provide a step-by-step guide through the process of designing an installation.

Chapter	Description
Chapter 1 Introduction	Includes description of InstallShield Express, the installation procedure, style and formatting conventions, an Express documentation overview, and an introduction to Express’ user interface.

Chapter	Description
Chapter 2 A Quick Tutorial	Provides a series of step-by-step examples of how to create your own installations, ranging from the very simple to the more complex.
Chapter 3 Building Your Installation	Addresses the following topics: setup planning, target and source directories, Express directory specifiers, and the Windows 95 logo requirements.
Chapter 4 InstallShield Objects	Describes the different objects available for Visual Basic 4.0, Visual C++ 4.1, Delphi 2.0, Borland C++ 5.0, and Paradox 7.0.
Chapter 5 InstallShield Extensions	Describes how to utilize extensions to call external functions during an installation.
Chapter 6 The Registry	Provides a quick look at the registry and how Express enables you to write your software's settings to your customer's registry.
Chapter 7 Frequently Asked Questions	Provides a list of common questions that Express users have experienced.
Chapter 8 Express Dialog Boxes	Contains a complete listing of every Express dialog box, including illustrations and field descriptions.
Appendix A Glossary	Lists the key terms used in this document, and the corresponding definitions.
Appendix B InstallShield Express License Agreement	Contains the InstallShield Express license agreement.
Appendix C Disk Builder Warning Messages	Lists the warning and error messages displayed by Disk Builder, and accompanying corrective steps.

Chapter	Description
Appendix D Index	Provides an alphabetical listing of topics addressed in the InstallShield Express Professional User's Guide.

InstallShield Express Documentation

The Express Help system and the Readme file are excellent sources of additional InstallShield Express information. Both of these documents feature full-color multimedia formatting, hypertext linking, and full-text search capabilities.

InstallShield Express Help System

InstallShield Express features a full-function online reference.

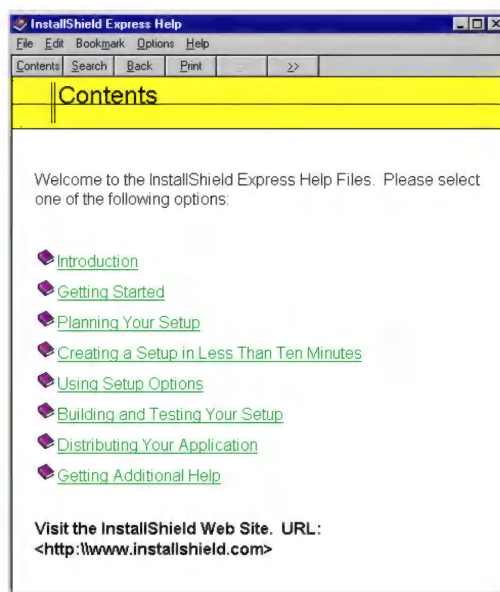


Figure 1.2. The Online Help Contents Screen.

Please take a few minutes and familiarize yourself with the Express help system. You can easily locate information by any of the following methods:

- Double-clicking on the book icons in the Contents window opens up individual topic icons. Double-clicking any of the individual topic icons displays the information you want.
- Clicking on the **Search** button located on the tool bar displays an alphabetical list of InstallShield Express topic titles. You can either select a topic from the list displayed or search for any entry you type into the field.
- Once inside the document, you can access any topic highlighted in green by simply double-clicking on the green text.

The online help provides instant access to points of interest, as well as a step-by-step guide to creating a custom installation.

Readme File

The InstallShield Express Readme file contains important information—vital, late-breaking news for InstallShield Express users. The latest updates about InstallShield Express are located in the Readme file, including known problems, documentation errata, and question-and-answer documents (FAQs).



Figure 1.3. The Readme Main screen.

The Express Readme file is where you can find the following information:

- The Readme topic provides information about the InstallShield family of products.
- Completion of the InstallShield Express product registration form ensures that you are eligible for the benefits available to registered users, such as special offers, updates, and exclusive InstallShield Express information. By registering, you can also access our superior technical support staff free for thirty days to help you with any Express questions you might encounter.
- A complete copy of the InstallShield Express license agreement is provided in the Readme file.
- A customer survey is included in the Readme file. Please take the time to complete and return this survey. We want to design the tools that you want to use. This survey is a way for you to tell us what Express functions you would like to see added or modified in the future. Your comments are very important to InstallShield Corporation's goal of providing you with the products, documentation, and support you have come to expect from InstallShield.
- We are committed to guaranteeing that our customers receive the best possible technical support. A technical support request form is included in this topic. To ensure that you receive the help you need in the quickest manner possible, please take a moment and fill out the request form.

User Interface

Now that we have explained the documentation, let's take a look at Express' user interface itself. Express features all of the Windows 95 functionality that you would expect while adding a point-and-click interface. This section takes a closer look at the sections of the basic Express window.

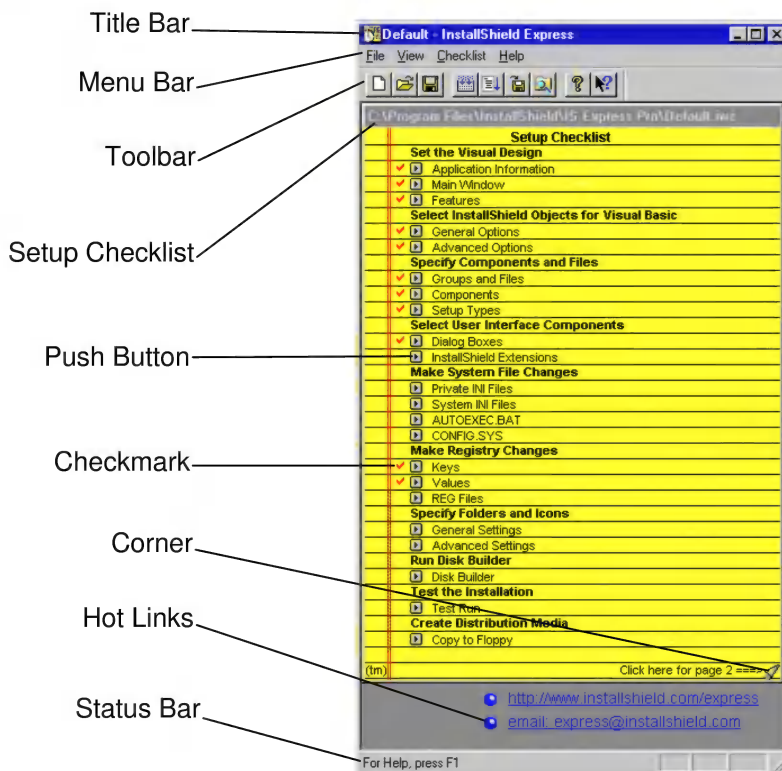



Figure 1.4. The Express User Interface.

The Express Checklist

The focal point of Express's interface is the Setup Checklist. Express provides you with a checklist to guide you through the process, ensuring that you include all necessary features—from displaying the correct on-screen logo all the way through to producing the correct installation disks to be sent to your customers.

The Express Setup Checklist is a sheet of steps needed to complete an installation. Each of the necessary steps to creating your installation is included in the checklist. All you need to do is to complete each of the sections in bold type according to the setup you have planned. Clicking the  icon next to each step opens the appropriate dialog.

As you finish each task, a red checkmark appears next to each completed item.

The Title Bar

The title bar displays your project's path and filename next to the full Express name. You can access the System menu by either left-clicking on the small icon at top left, or right-clicking anywhere on the bar.

The Menu Bar










The Express menu bar is always visible within Express. It contains four drop-down menus:

- The **File** menu contains New, Open, Close, Save, Save As, and Exit options, as well as the most recently used files and shortcut keys.
- The **View** menu allows you to display or hide the toolbar and status bar.
- The **Checklist** menu provides an alternative method for accessing Setup Checklist items using a mouse or your keyboard.
- The **Help** menu gives you quick access to the Quick Tour, Help Index, Read Me File, Online Registration, Technical Support Request Form, version information, and the InstallShield Express web site.

The Toolbar





Many of the essential InstallShield Express tools are included on the toolbar, which is located at the top of the main Express window. While some of these buttons should be very familiar to you from other Windows applications, others will be new to you.

Note The toolbar can be hidden by deselecting it from the View menu.

Button	Name	Function
	New	Opens the New Project dialog box, which allows you to start a new Express project.
	Open	Opens the Open dialog box, which enables you to open an existing Express project file.
	Save	Automatically saves your project under its current name and location.
	Build	Launches the Disk Builder, which creates a data file from your settings, compresses your application and installation files, splits the compressed files, and places the split files into separate directories which correspond to your distribution disks.
	Test Run	Executes a test run of your setup on your own machine.
	Copy to Floppy	Launches the Copy to Floppy function, which transfers the disk images created with the Disk Builder to the distribution media you select.
	Explorer	Launches Windows Explorer, which you can use to drag and drop files into an open dialog box.
	About	Opens the About InstallShield Express message box, displaying program and copyright information and the version number.
	Help	Accesses context-sensitive help. When you click on this push button, the question mark icon will appear to be “attached” to your arrow icon. Point the arrow at the checklist item for which you want help, and single-click to open Express Help on that particular topic.

Other Symbols

Express utilizes various on-screen aids to help speed you through the building of your installation. These symbols are used consistently throughout Express to indicate what your next action should be. The following table lists all of the visual aids used in Express:

Symbol	Name	Function
	Check marks	Checkmarks serve as reminders. A checkmark appears next to each dialog in which you have made and accepted settings. You do not need to have all, or any particular number, of the dialogs checked in order to create your setup.
	Push Buttons	You can instantly access any of the Express dialogs by clicking on the appropriate push button or the name of the dialog.
	Corner	On lower resolution monitors, click the bent corner to reveal the rest of the Setup Checklist. On all resolutions, clicking the bent corner reveals additional InstallShield Express information.
	Status Bar	Displays basic information about the control underneath the arrow. The status bar can be hidden by deselecting it from the View menu.

In addition, the background of the main Express window displays both the InstallShield Express email address and Internet website location. Express enables you, given your machine has the necessary capacity, to contact the InstallShield Express World Wide website by simply clicking the appropriate link. Please feel free to contact us with your comments and suggestions about Express.

Ideas for Improvements

Our goal is to enable you to quickly create bulletproof installations. If you have a suggestion for a future edition of InstallShield Express or a feature you would like to see added in an upcoming release, please let us know. We have created a special email address for your suggestions. Send your idea to us at isxwish@installshield.com.

Chapter 2

A Quick Tutorial

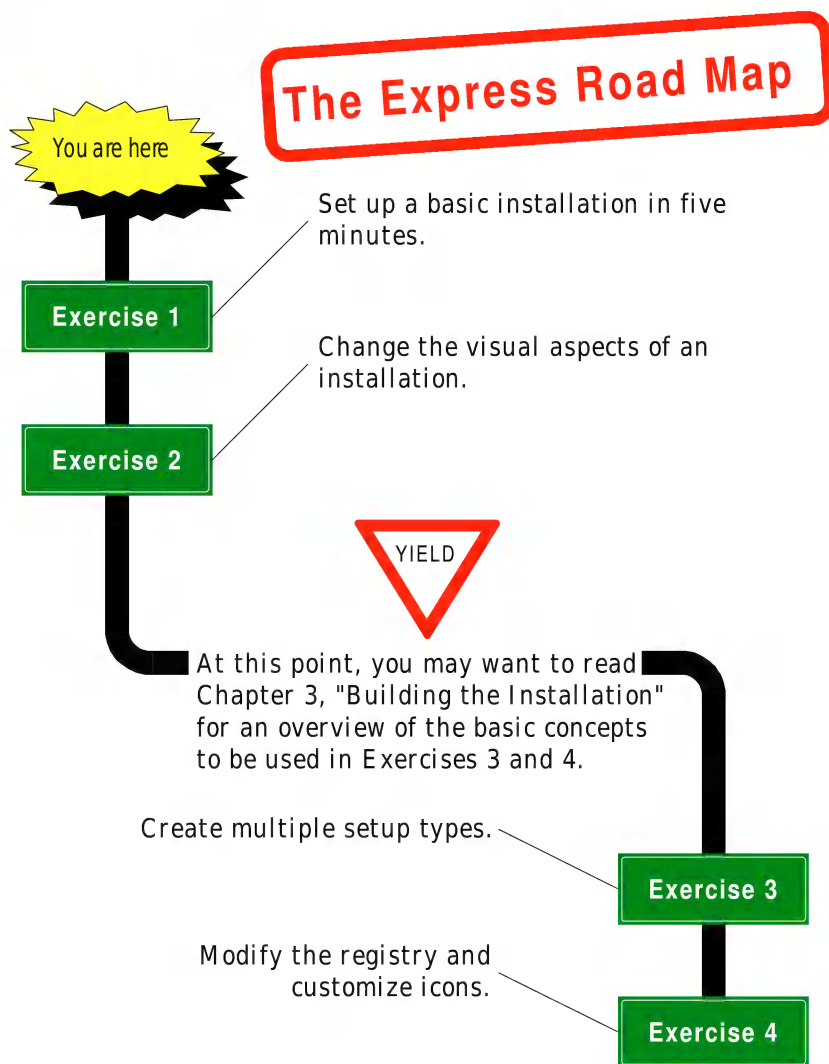
With InstallShield Express, you can create your own custom installation program in under five minutes. This chapter walks you through examples of different tasks you can perform in creating your installation, starting with a very basic project and increasing in difficulty with each exercise. In case you do run into a problem, remember that all Express dialogs have context-sensitive help topics which can be accessed at any time by pressing F1 or clicking the Help button.

Contents

Getting in the Express Lane	2-2
Exercise 1	
Creating an Installation in Under Five Minutes.....	2-3
Exercise 2	
Creating a Unique Look for the Installation.....	2-11
Exercise 3	
Adding Multiple Setup Types to the Installation	2-20
Exercise 4	
Modifying the Customer's Registry and Adding Customized Icons.....	2-29

Getting in the Express Lane

In the previous chapters, you may have read our claims that InstallShield Express is simple. Instead of taking our word for it, why not try it yourself? We believe that the best way to understand Express is to jump right in and use it.



Exercise 1

Creating an Installation in Under Five Minutes

Let's start by creating a basic installation. When you finish this exercise, you will have seen exactly how simple it is to create a setup using Express. This exercise illustrates the following procedures:

- Opening a new project.
- Entering information about your application.
- Selecting the main .EXE file.
- Building the setup.
- Testing your installation.

For this exercise, we will create a sample setup that will install an application you should already have on your system, the Windows calculator. For this exercise, simply follow the steps provided and you will quickly create a working installation and begin to understand how the entire process comes together.

1. Launch InstallShield Express by either double-clicking the  icon or selecting it from the Windows 95 Start menu. The following **Welcome** dialog will be displayed.

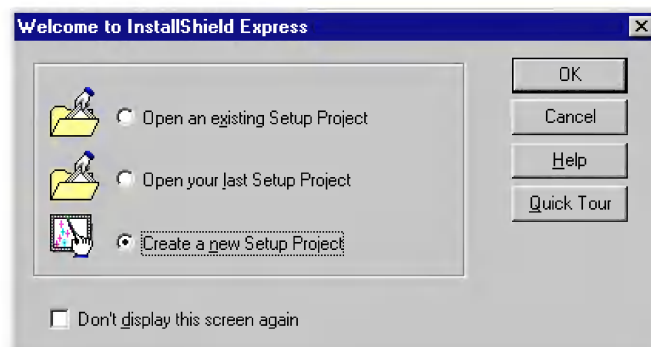



Figure 2.1. Welcome to InstallShield Express dialog.

If you do not see this dialog box, select **New** from the **File** menu or click the  icon to create a new project and move to step 3.

2. Since you are creating a new installation, select **Create a new Setup Project**, then click the **OK** button.

Note If you do not want this dialog box to be displayed when you open Express, select the **Don't display this screen again** checkbox before you select OK. Selecting the checkbox causes this dialog box to be permanently hidden from this point forward. If you select this option, all of the functions appearing in this dialog box are always available in the Express menu bar.

3. Express will then launch the **New Project** dialog box.

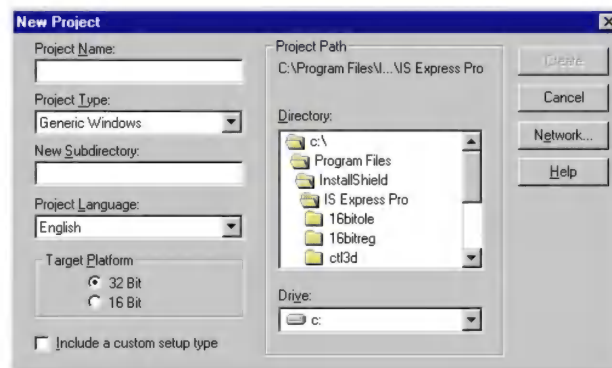


Figure 2.2. The New Project dialog box.

4. Select the drive and directory where you want to store this project using the standard **D**irectory and **D**rive controls. The current path for the project file is displayed at the top of the box.
5. If you want to create a new subdirectory for your project file, simply enter its name in the **New Subdirectory** field. Express will automatically create the directory under the path you specified in step 4 and place your project file in it.
6. Enter the name of your project in the **Project Name** field. Express will use this name, followed by the .IWZ extension, as the filename for this project. For this example, type `YourProject` in this field.

7. Select the desired development environment from the **Project Type** drop-down menu. This selection will determine which set of InstallShield Objects will be included with your setup project. If you are not using any of the platforms supported by Express, you can select **Generic Windows**.
8. In the **Project Language** field, you can specify the language you want your installation to be created in. The dialogs displayed to your customers during the installation process can be in English, French, or German. More languages will be added in the near future.
9. Select the **16 Bit** or **32 Bit** radio button to specify the target platform for your setup.

Note If the development environment selected in the **Project Type** field does not support both 16-bit and 32-bit settings, these radio buttons will not appear in the **New Project** dialog box.

If you are uncertain as to which setting you should select for your project, refer to Chapter 3, "Building Your Installation."

Note We recommend that you create both a 16-bit installation and a 32-bit installation if you plan to have your application installed on both 16-bit and 32-bit platforms. However, if you are only going to create one installation to cover both target platforms in this situation, we suggest using the 16-bit target platform setting. Note that you will not have access to 32-bit InstallShield Objects if you select the 16-bit setting.

10. When you are satisfied with your entries, click the **Create** push button. The **New Project** dialog box closes and the Setup Checklist is displayed.

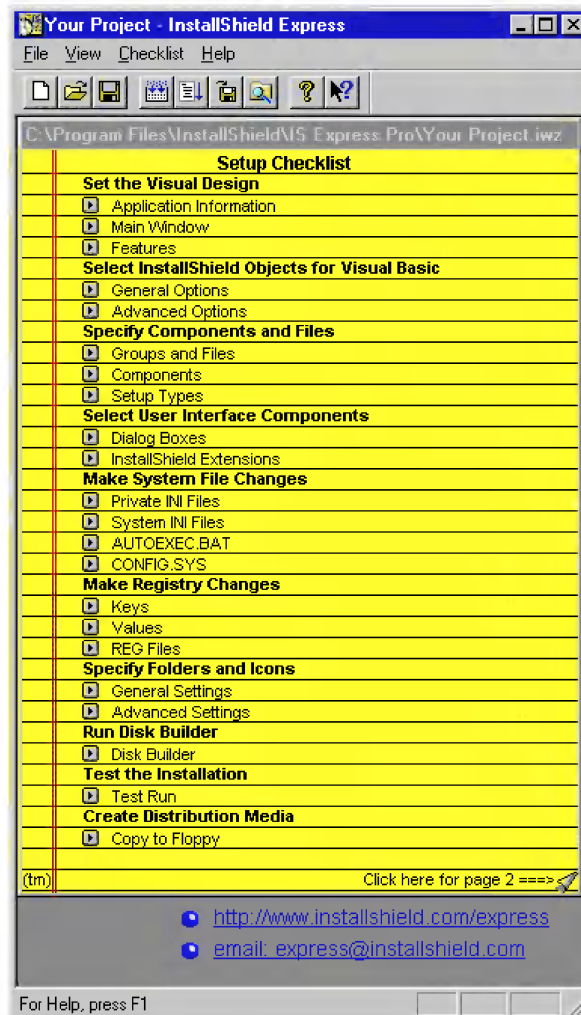


Figure 2.3. The Setup Checklist.

You will notice that the file location information entered in the **New Project** dialog box appears in both the title bar of the window and the top line of the Setup Checklist.

11. Click the **Application Information** heading to display the **App Info** tab of the **Set the Visual Design** dialog box.

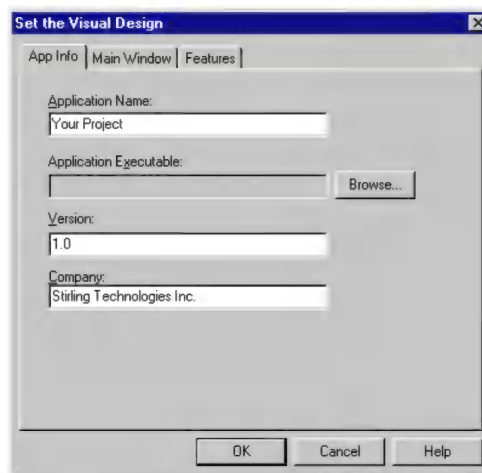


Figure 2.4. Application Information tab.

12. Notice that default entries appear in the **Application Name**, **Version**, and **Company** fields. The **Application Name** setting is the project name you entered in the **New Project** dialog box. The default **Version** setting is always 1.0, but will update automatically based on the application executable you select in step 14 below. The **Company** default is read from your system's registry. All of these fields can be modified by simply typing the desired information in the appropriate field.

Note Do not leave any of these three fields blank, or Express will not be able to create the automatic registry entries necessary to comply with Windows 95 Setup Guidelines. Use the default settings or manually change the settings.

13. Click the **Browse...** push button next to the **Application Executable** field to launch the **Open** dialog box. Find and select the main executable file of your application to install—in Windows 95, you can use **CALC.EXE**, located in your main Windows directory, for this exercise.
14. When you click the **Open** push button to select your .EXE, the filename will appear in the **Application Executable** field, preceded by the [Program Files] group name directory specifier.

Note For more information on InstallShield Express' directory specifiers, refer to Chapter 3, "Building Your Installation."

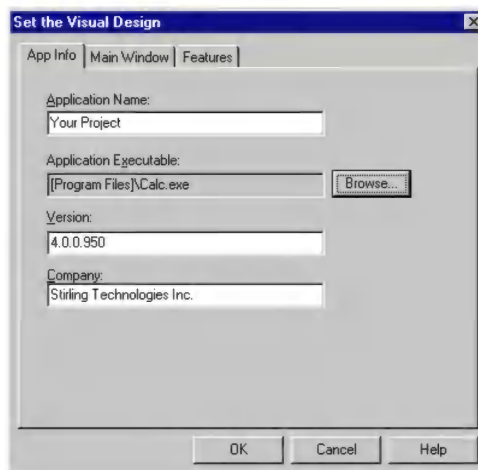


Figure 2.5. Selecting the Application Executable.

15. To apply the changes made in the **Set Visual Design** dialog boxes, click the **OK** button. The Setup Checklist reappears with a check mark next to the Application Information item.

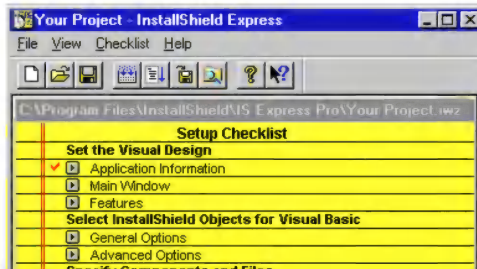


Figure 2.6. The Setup Checklist after modifying Application Information.

16. Now build the installation. Open the **Disk Builder** dialog box, located near the bottom of the Setup Checklist. (On VGA resolution, the **Disk Builder** item may be on the second page, accessed by clicking the bent corner in the lower right-hand corner of the "notepad.")

Note The actual copying of your files to the distribution disks does not take place at this point. In this step, only the disk images are created. You still have opportunities to test and correct your application before actually copying the images to disk.

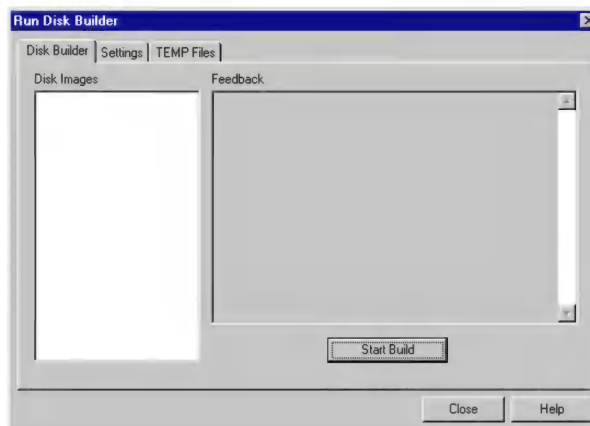


Figure 2.7. The Run Disk Builder dialog box.

18. Click the **Start Build** push button to start the process. The cursor will change into an hourglass and a progress indicator window will appear while the disk images are built. Status messages are displayed in the Feedback window to keep you informed.

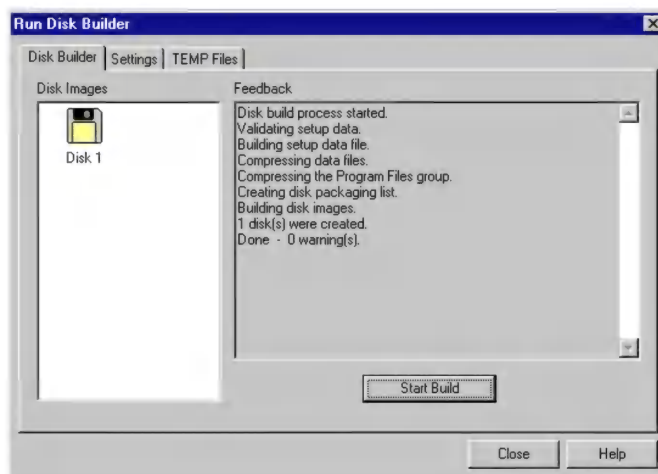




Figure 2.8. A completed disk build.

If you should receive a warning message during the disk build, do not worry. The warning will not cause the Disk Builder to stop executing, but the condition that caused the warning may cause serious problems in your setup. Most errors encountered during this disk build are very

simple to fix. To assist you, Express contains a complete listing of warning messages in the online help.

Click the **H**e**lp** button in this dialog box, then the  icon at the bottom of the Run Disk Builder to access the Disk Builder Error Message topic. You can view a listing of warnings which include the warning number, description, and a possible solution for the problem. This information also appears Appendix C, "Disk Builder Warning Messages."

19. Click the **C****lose** push button to return to the Setup Checklist.
20. Test the disk image you just built. Express enables you to check the performance by clicking the  next to the **T****est the I****nstallation** entry on the Setup Checklist. This launches an actual installation of your application on your system. All of the billboards, dialog boxes, and graphics that your customers will see appear on your monitor. This enables you to preview your installation before committing it to disk.

That's how simple it is to create a very basic installation using Express. Note the professional look and feel of the setup you have created. Note also that Express automatically creates the icon and places it on the Start menu.

Customizing the installation to display your company's logo or other information is just as easy. The next exercise shows you how to create your own look for your software's installation.

Exercise 2

Creating a Unique Look for the Installation

This exercise illustrates the following procedures:

- Opening an existing project.
- Modifying the window background and text.
- Adding a logo bitmap.
- Creating a new file group for .DLLs to be installed in the System directory.
- Adding a Readme file to the setup.

As a developer, you are well aware of the complexity involved in installing your application on a customer's system. Directories must be created, registry keys added or modified, and compressed files must be decompressed and copied onto the target system. Of course, most customers are unaware of how much activity is going on in the background because all that appears on their monitors is your installation's main window.

This exercise shows you how to customize the main window of your installation. The following graphic illustrates a sample setup appearance and identifies certain elements which will be explained in this lesson.

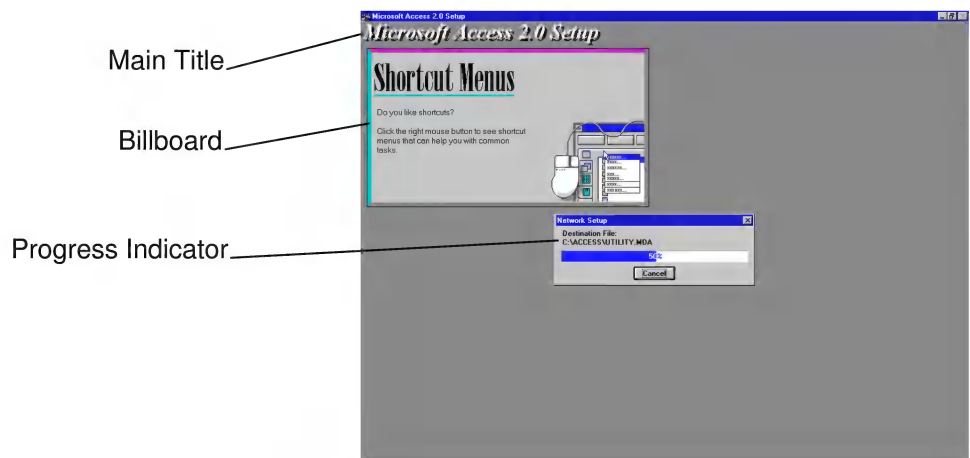



Figure 2.9. A sample setup main window.

1. Open the project you worked on in Exercise 1 by selecting **Open...** from the **File** menu and browsing for the file, or by selecting the file from the most-recent list at the bottom of the **File** menu. The Setup Checklist appears.
2. Click the  next to the **Main Window** item in the Setup Checklist. The **Main Window** tab of **Set the Visual Design** dialog box opens.

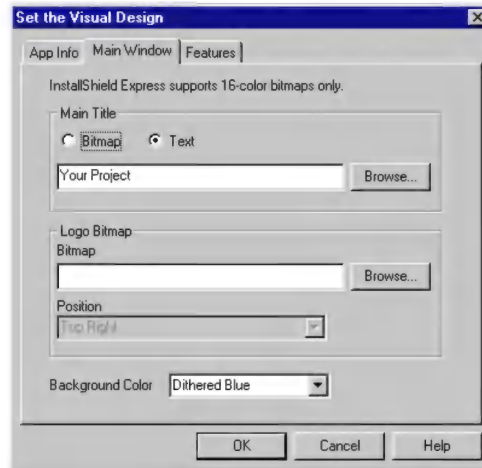


Figure 2.10. The Main Window tab.


The **Main Window** dialog enables you to quickly modify the visual design of your installation. You can display a text or bitmap title across the top of the window, as well as a logo located anywhere on the window you want.

3. To enter a text main title, click in the field below the radio buttons and type the title text **EXACTLY** as you want it to appear. The text you enter must be no greater than eighty characters. In our example, we changed the main title from "YourProduct" to "Welcome to YourProduct."
4. To display a logo in the background during the installation, click the **Browse...** button to search for the bitmap file you want to use. You can also type the complete path and filename in the **Bitmap** field. If you do not have a logo in a bitmap to display, select any small 16 color bitmap image on your system. We just want you to get a feel for how to manipulate these images in Express.

Note InstallShield Express supports any 16 color (or less) bitmap image (.BMP) or placeable Windows metafile (.WMF). However, the image you select cannot be RLE encoded. Express will not support this type of compression.

5. Once the filename is entered into the **Bitmap** field, you can specify the position of the image in the main window during installation by selecting one of the choices in the **Position** drop-down combo box.

Note If you select **Top Left** for your logo bitmap, and you also entered text or a bitmap in the **Main Title** window, the logo bitmap will be displayed behind the main title text or bitmap.

7. Select the background color of your main window from the **Background Color** combo box. Express offers both solid and dithered options.
8. After you have made all of your selections, click the **OK** push button to apply your changes or **Cancel** to close the dialog box without saving the changes.
9. Click the  next to the **Groups and Files** item on the Setup Checklist. This opens the **Groups** tab of the **Specify Components and Files** dialog box.

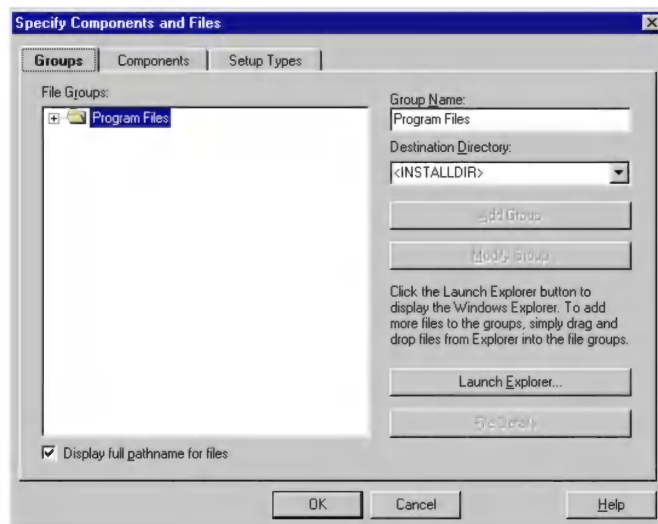


Figure 2.11. The Groups tab.

10. Add a new group for the .DLLs required by your application. In the **Group Name** field, type a name for the new group, such as `DLL Files`.
11. Express enables you to specify where these files will be copied on the target system through the use of Express Directory Specifiers. In our example, we want all of our .DLLs to be installed in the target `WINDOWS\SYSTEM` directory. To specify the `SYSTEM` directory, select **<WINSYSDIR>** from the **Destination Directory** drop-down list.
12. To add the new group, click the **Add Group** push button. The dialog box will update to display the `DLL Files` group along with the existing `Program Files` group.
13. Add the .DLL files to the `DLL Files` group. Click the **Launch Explorer...** push button to launch or restore Windows Explorer. (You can also use File Manager for this task.) Position your windows such that you can see the files in Explorer and the group names in Express. In Explorer, highlight the .DLL files you want to copy. Drag and drop them onto the `DLL Files` group in the **Groups and Files** dialog box.

Once a file has been added to a group, you cannot simply reassign it to another group. If you need to place a file (or files) into a different group, delete it from its current group listing and then add it to the other group.

Note When you add files to a group, InstallShield Express copies the full path and filename of each file. Therefore, do not change the path or filenames of any files after dragging and dropping them onto the groups, or else the Disk Builder will be unable to locate your files when it creates your setup.

14. To accept the changes made in the **Specify Components and Files** dialog box, click the **OK** push button to enter your changes, or **Cancel** to close the dialog box without making any changes.
15. Express enables you to select the dialogs that your customer views during the installation. Click the **Dialog Boxes** push button in the Setup Checklist to open the **Select User Interface Components** dialog box.

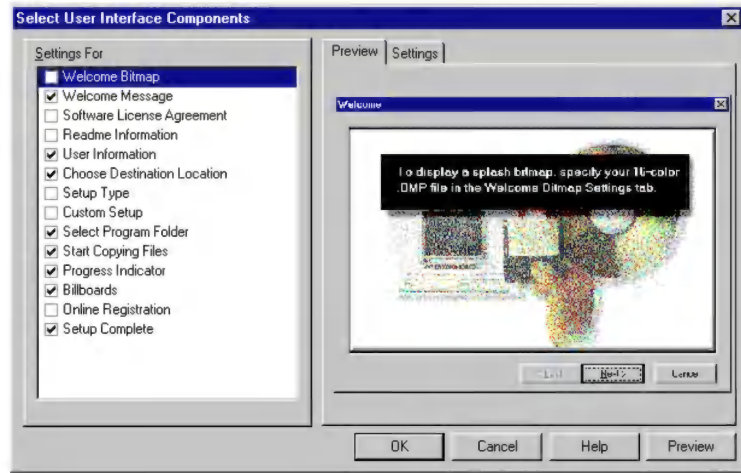


Figure 2.12. Dialog Boxes dialog box.

For specific information on the user dialogs, refer to the following table:

Dialog Box	Overview
Welcome Bitmap	Allows you to display a bitmap in a child window after the Startup Message, but before any of the other user dialogs.
Welcome Message	Displays a message welcoming your customer to your application's installation, instructions to exit all Windows programs before starting the setup, and a brief copyright warning.
Software License Agreement	Allows you to display a text file containing a copy of your license agreement.
Readme Information	Allows you to display a Readme text file to the customer before the setup begins collecting information.
User Information	Collects the customer's name, company, and, optionally, serial number. This information is then automatically stored in the registry (for 32-bit installations) or WIN.INI (for 16-bit installations).

Dialog Box	Overview
Choose Destination Location	Allows your customer to select the main installation directory for your application on his or her system. The directory and path selected will replace the <INSTALLDIR> directory specifier in any of the Express dialog boxes in which you have used it.
Setup Type	Allows the customer to select a Typical, Compact, or Custom installation. You must create the components for each of these setup types in the Specify Components and Files Express dialog box.
Custom Setup	Is displayed only if the customer selects a Custom setup type in the Setup Type user dialog box. The Custom Setup dialog box allows your customer to select which components of your application he or she wishes to install.
Select Program Folder	Allows the customer to choose the name of the program folder (Windows 95) or group (Windows 3.1 or NT) in which your application icons will be placed.
Start Copying Files	Displays the setup type, destination directory, and customer information selected by your customer. It allows him or her to double-check the selections and, if necessary, go back and modify one or more of them.
Progress Indicator	Gives the customer a graphical representation of the file-transfer process. The progress bar gradually increments to depict the percentage of the file transfer process which has been completed, while the text field above the bar displays the names of the files as they are copied.
Billboards	Allows you to display billboards during your installation. You can use bitmap (.BMP) files or Windows metafiles (.WMF) as your billboards.

Dialog Box

Overview**Online
Registration**

InstallShield Express includes an Online Registration feature which enables your customers to register their copy of your product during the installation. In order to activate this feature, you must contact Pipeline Communications, Inc. at 1-800-WIN95REG, or by email at WIN95REG@pcpipeline.com to set up your service. You can also contact us here at InstallShield by telephone at (800) 374-4353 or email us at info@installshield.com for more information.

**Setup
Complete**

The **Setup Complete** user dialog actually consists of two dialog boxes: a reboot computer dialog box and a launch application dialog box. The preview shows only the **Reboot Computer** dialog box.

16. To see how easy it is to modify a component, let's add a Readme file to our installation. The **Readme Information** user dialog box allows you to display a text file before the setup begins collecting information. Click the check box next to the **Readme Information** entry to add this dialog to your setup. Clicking the **Preview** push button gives you a closer look at the dialog.

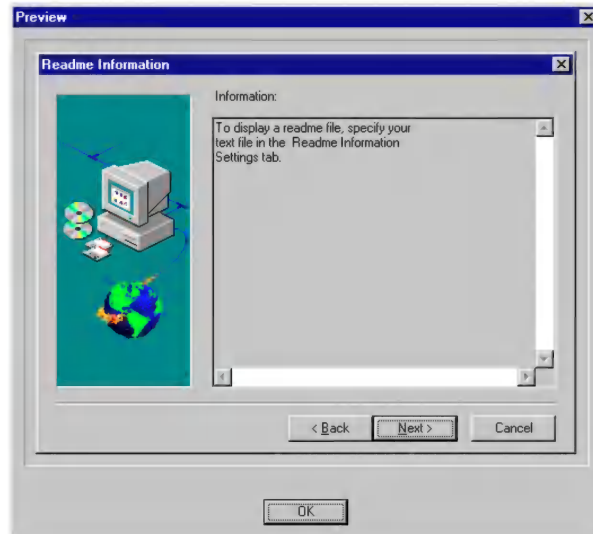


Figure 2.13. Readme Information preview box.


17. The actual text that appears in the Information window is pulled from a text file that you can create in any text editor.

Note The contents of the file will not be automatically formatted to fit the width of the edit field in which the **Readme Information** dialog box displays the text. If you do not want the customer to have to use the horizontal scroll arrows to read it, you must format your text file using hard returns at the end of each line to be displayed.

Specify the text file in the **Settings** tab of the **Readme Information** dialog box either by using the **Browse...** push button to search for the file on your system, or by typing the fully qualified path and filename in the edit field.

Note The Readme Information preview is a sample image. The preview image will not change to reflect your specified file.

18. To accept the changes you made in the **Select User Interface Components** dialog box, click the **OK** push button. Click **Cancel** to close the dialog box without making any changes.

19. Now rebuild the installation with the changes we have made in this exercise. Click the  next to the **Disk Builder** entry and follow the instructions from step 16 of Exercise 1.

Congratulations! Now you know how to create an installation customized to your visual specifications. You also know how easy it is to add a new group to your installation.

So far, we have dealt with an installation which includes a single setup type (Complete). The next exercise shows you how easy it is to design your own multiple setup types, which will give your installation the flexibility to fit a wide range of potential customers.


Exercise 3

Adding Multiple Setup Types to the Installation

This exercise illustrates the following procedures:

- Creating a new project.
- Creating multiple setup types.
- Adding and modifying groups.
- Creating setup components.
- Adding groups to components.
- Adding components to setup types.

If you have not already, you may want to read the Planning the Installation section in Chapter 3, “Building Your Installation.” Many of the concepts used in this exercise are discussed in greater detail in that section.

1. Select **N**ew from the **F**ile menu or click the  icon to open the **New Project** dialog box.

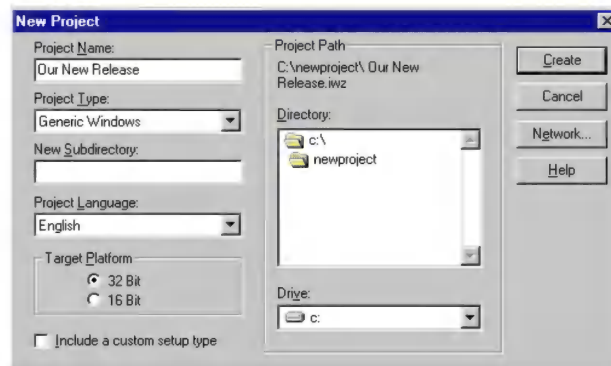


Figure 2.14. The New Project dialog.

In this dialog, make the necessary entries to create a new setup, such as that shown in Figure 2.14. Notice the **Include a custom setup type** check box at the lower left of the dialog box. Selecting this option enables

you to establish different setup options for your installation. For a quick refresher on creating a new project, refer to Exercise 1 of this chapter.

2. Select the **Include a custom setup type** check box and click the **Create** push button.
3. Open the **Application Information** tab of the **Set the Visual Design** dialog and select your executable file. For the purposes of this exercise, you can use WORDPAD.EXE, the Windows 95 native word processing program, which is typically found in the /Program Files/Accessories directory of your Windows drive. After selecting the .EXE file, click **OK**.
4. Let's set up the file groups used to create the different setup types. Open the **Groups** tab of the **Components and Files** dialog by clicking the push button next to **Groups and Files** in the Setup Checklist.

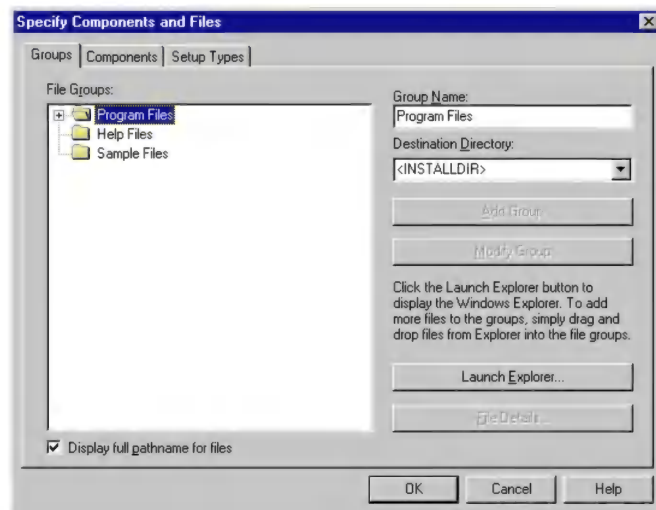


Figure 2.15. The Default file groups.

Express's default groups appear in the **File Groups** window. You can modify or delete these groups, as well as add new groups as required. The settings for the highlighted group appear to the right of the window. In our example, we will use three file groups, Documentation Files, Program Files, and Converter Files, to build our setup types.

5. To change the name of the Help Files group to Documentation Files, first highlight **Help Files** in the **File Groups** window. In the **Group Name** field, enter the new name for the group. To accept the new name, click the **Modify Group** push button. The **File Groups** window will update to display the renamed group.
6. We do not need a Sample Files groups, but we do need a group to house the conversion utilities that a customer might want to install with WordPad. Highlight **Sample Files** in the File Groups window and replace the entry in the **Group Name** field with Converter Files. At this point, the dialog box will look like this:

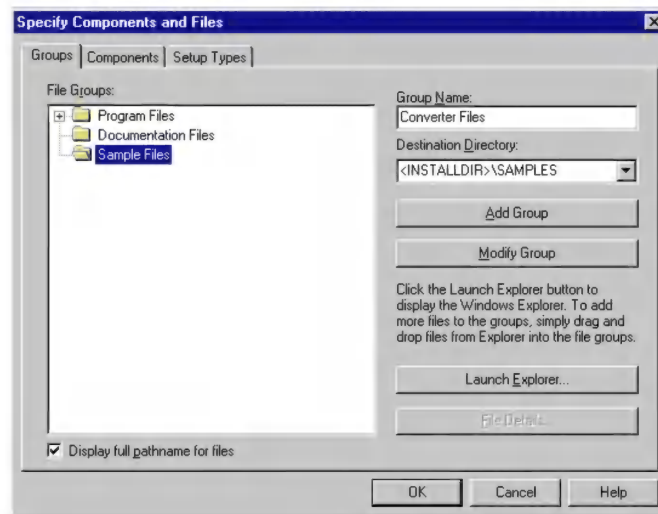


Figure 2.16. Modifying group names and directories.

7. Note the entry in the **Destination Directory** field. Although we changed the name of the file group, the old destination directory remains. For the sake of example, let's assume that WordPad requires MSWD6_32.WPC and WRITE32.WPC to be in a \CONVERTER subdirectory. If this were the case, these conversion files would not run properly if they resided in the \SAMPLES subdirectory on the target system. Change the entry in the **Destination Directory** field to <INSTALLDIR>\CONVERTER. To apply your changes, click the **Modify Group** push button.

Note If you click the **OK** push button before clicking the **Modify Group** push button, the changes you have made in this dialog will not be saved.

8. The next step is to add files to the groups that we have created. Clicking the **Launch Explorer...** push button and add files by selecting the files in Explorer and dragging them to the correct file group in the File Groups window. After adding files to their groups, the File Groups window might look like this:

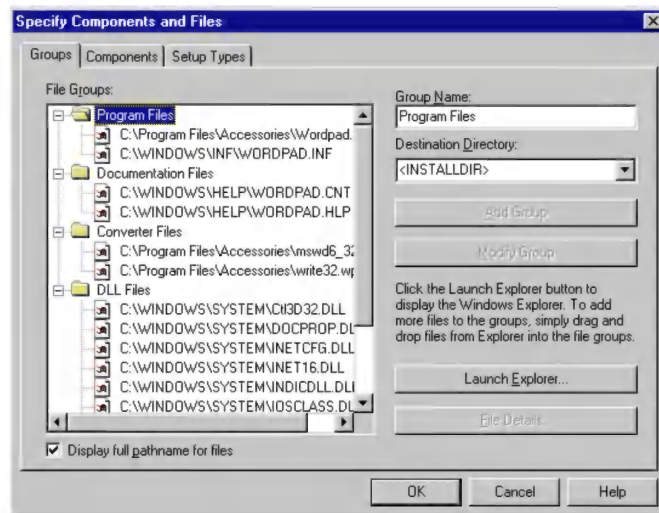


Figure 2.17. File groups with files added.

9. To view information regarding any of the files displayed in the File Groups window, highlight the file and click the **File Details...** push button. This reveals the file's date, size, and version number.
10. Now that the file groups are set, you must assign them to components. Click the **Components** tab of the **Specify Components and Files** dialog box.

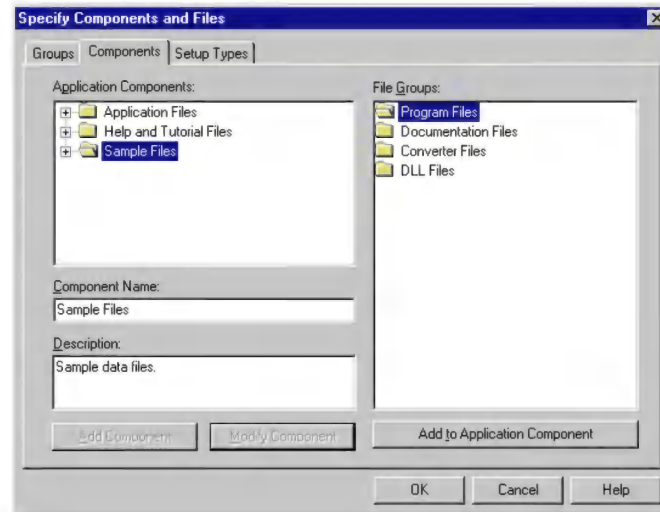


Figure 2.18. The default components.

The **Application Components** window displays Express's default components. The name of the component and a short description of the highlighted component's contents appear in the **Component Name** and **Description** fields. This information is presented to the customer during an installation in the **Custom Setup (Choose Options)** dialog box. To add a new component, change the information in either of these fields and click the **Add Component** push button.

11. One of the default components is named Sample Files. Since our application does not include any sample files, this component is unnecessary. The conversion files could be considered an optional component which a customer might not want to install, so let's place them in a component of their own. With the Sample Files component highlighted, change the entry in the **Component Name** field to `Conversion Utilities`. In the **Description** field, enter a new description (no more than 150 characters) for this component. To save your changes, click the **Modify Component** push button. The **Components** tab will update to look something like this:

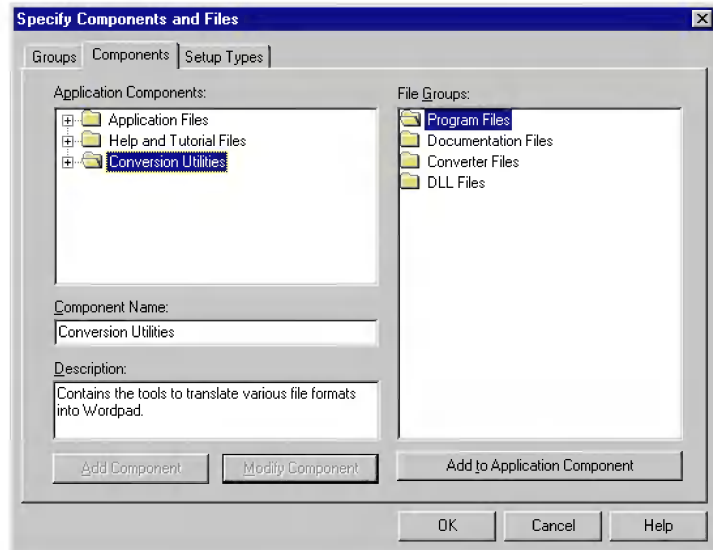


Figure 2.19. Modifying setup components.

Note If you click the **OK** push button before clicking the **Modify Component** push button, the changes you have made in this dialog will not be saved.

12. Once you have modified the component names and descriptions, you must assign file groups to each component. Keep in mind that each of these components can be individually selected by the customer during a custom installation. Therefore, you should place similar file groups in the same component. Express, by default, assigns certain file groups to specific components.

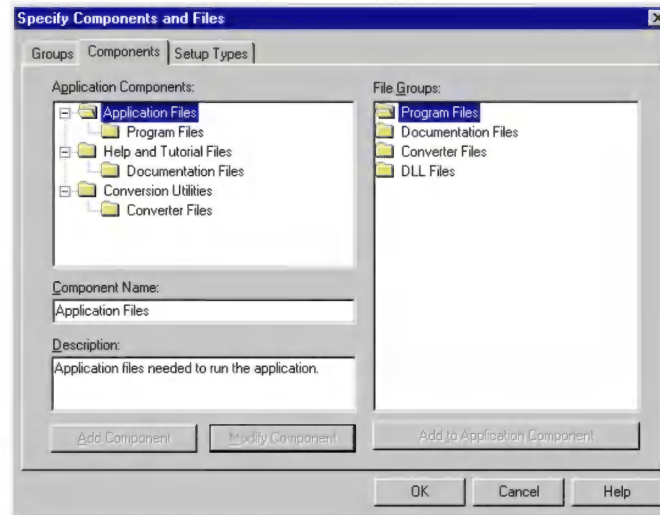


Figure 2.20. Default File Group Assignments.

As you can see in Figure 2.20, three of the four file groups in this example are already assigned to components. Let's add the DLL Files group to the Application Files component. To add a file group to a component, simply highlight both the file group to be added and the target component and click the **Add to Application Component** push button. In our example, the **Components** tab will look like this after adding the DLL Files group.

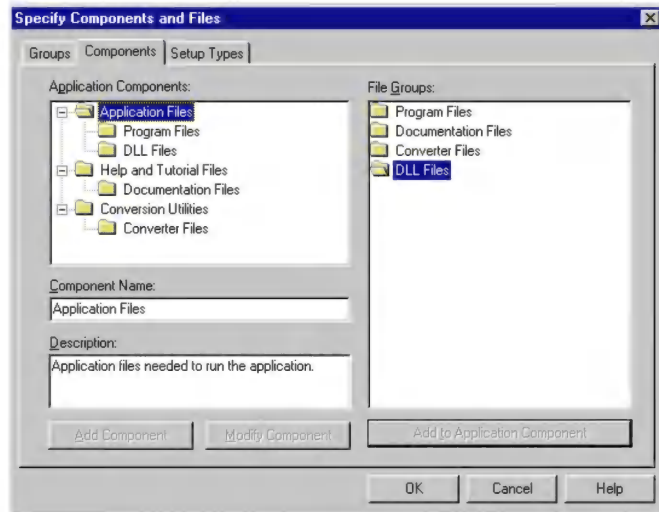


Figure 2.21. Modified application components.

13. The last task in the process is to assign your components to the different setup types. Click the **Setup Types** tab to display the Setup Types and Application Components windows.

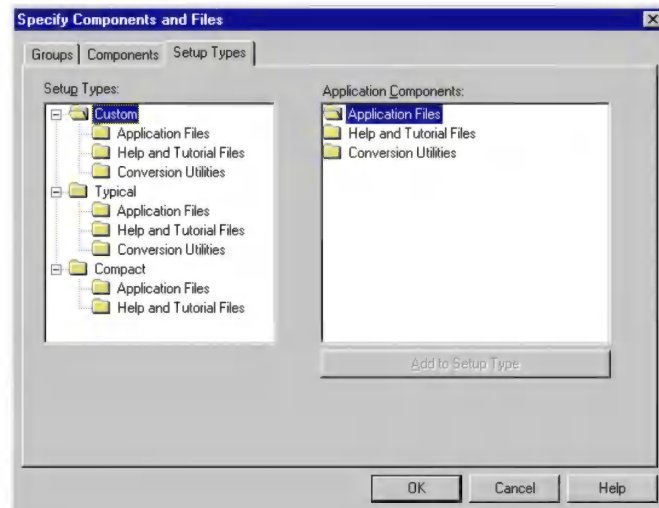



Figure 2.22. The Setup Types tab.

14. Express assigns certain components to certain setup types automatically. All of your components are by default included in the Typical and

Custom setup types. To remove a component from a setup type, highlight the appropriate component in the Setup Type window and press the Delete key. For a Compact Setup, include only those components necessary for minimal operation of your application. To add a component to a setup type, highlight it in the **Application Components** window, highlight the setup type to receive it, and click the **Add to Setup Type** push button.

Note The only setup types available within Express are Typical, Custom, and Compact.

15. After the file groups, components, and setup types are configured, click the **OK** button to enter your selections.
16. Once the custom setup types are organized, build the installation in the **Disk Builder** dialog, as outlined in Exercise 1.
17. Now click the  push button next to **Test the Installation** in the Setup Checklist to launch the setup you built. Run a test using the Typical setup type. Note what files are copied and where. Uninstall the application (in Windows 95, use the Add/Remove Programs applet located in the Control Panel). Now test it using the Compact setup type. Run several tests using Custom setup types, varying your component selections and noting the results. Be sure you uninstall the application between each test.

The final exercise in this chapter walks you through tasks like modifying your customer's registry during the installation and adding customized icons to your customer's system.

Exercise 4


Modifying Your Customer's Registry and Adding Customized Icons

This exercise illustrates the following procedures:

- Opening an existing project.
- Adding an entry to an .INI file.
- Adding a key and a value to the registry.
- Creating a customized program icon.
- Building the distribution disks.

If you are unfamiliar with the basic concepts of the Windows 95 registry, you should read Chapter 6, "The Registry." Some of the concepts addressed in this exercise are explained in greater detail in that chapter.

Note Microsoft Windows 95 Logo Requirements recommend that your installation does not make changes to your customer's WIN.INI file. Information which would have been entered into the WIN.INI file in Windows 3.1 or NT should now be loaded into the registry or your private .INI files. This exercise makes changes to the WIN.INI file for demonstration purposes only

1. Open the project you worked on in Exercise 3 and click the  button next to the **System INI Files** item in the Setup Checklist to open the **Make System Files Changes** dialog box.

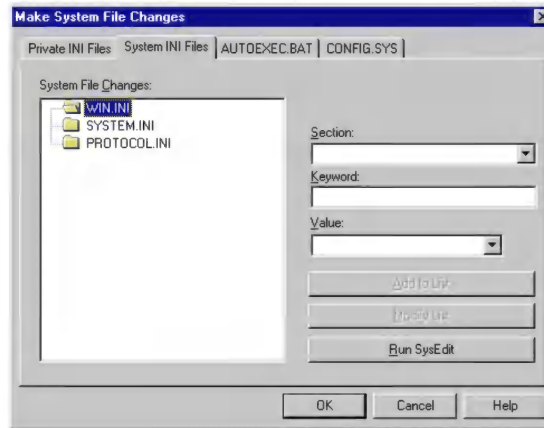


Figure 2.23. The System INI Files tab.

2. This dialog enables your installation to modify your customer's system .INI files. Other tabs in this dialog allow you to create entries during the installation in the target AUTOEXEC.BAT and CONFIG.SYS files, as well as private .INI files used by your application, after receiving the target system's setup information. For this exercise, we'll add an entry to the WIN.INI file on the target system that will launch WordPad when a .TXT file is selected. In the System File Changes window, click the WIN.INI file to highlight it.
3. In the **Section** field, you can type the section name or select it from the drop-down list, which will contain some of the more common section names in the highlighted file. In our example, select `Extensions`.
4. Enter the keyword (that part of an INI-file entry line that precedes the equal sign) in the **Keyword** field. Since we are adding a line to handle .TXT extensions, type `txt` in the field.
5. In the **Value** field, you specify the value, or that part of the entry that follows the equal sign. In order to take advantage of installation information, you can use Express Directory Specifiers in this box. In our example, we will enter `<INSTALLDIR>\WORDPAD.EXE ^.TXT`. This ensures that the system will be able to locate the executable, regardless of where the customer installs it.

6. To save the information for this new line, click the **Add to List** push button. The dialog box updates to display the added line.

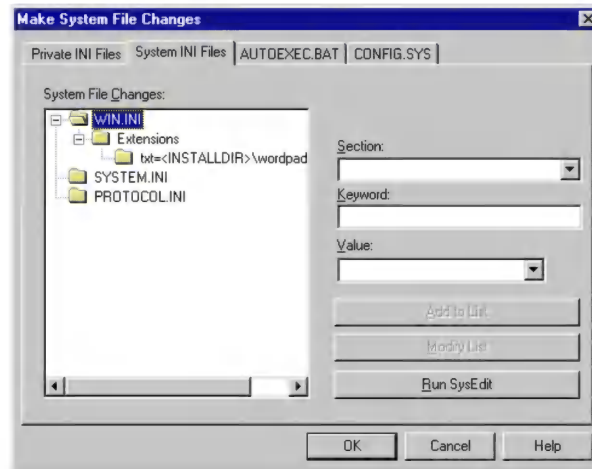


Figure 2.24. Displaying Added .INI Entry.

Note If you click the **OK** push button before clicking the **Add to List** push button, the changes you have made in this dialog will not be saved.

7. Click the **OK** push button to close this dialog.
8. Now let's create a Help File registry key for our example. Click the **Keys** push button on the Setup Checklist to open the **Make Registry Changes** dialog box.

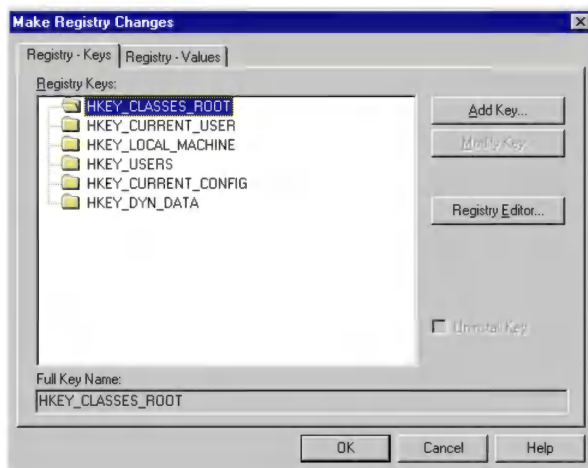


Figure 2.25. The Registry - Keys tab.

9. The Registry Keys window displays the six root keys for the Windows 95 registry. These keys cannot be modified. However, you can add subkeys underneath any of these existing keys. We will add a key under the HKEY_LOCAL_MACHINE root key by first highlighting HKEY_LOCAL_MACHINE in the Registry Keys window and clicking the **Add Key...** push button. This opens the **Registry New Key** dialog box.

Note If you are using Express to create a 16-bit installation to run under Windows 3.1, any registry keys that you create will appear under the HKEY_CLASSES_ROOT key, regardless of what you entered in the **Registry - Keys** dialog box.

10. In the **New Key** field, enter the name you want to give the key. You can enter more than one level of subkeys at once by using a backslash to separate each level, as shown in Figure 2.26. Application-specific registry keys that you add should conform to the following format: Software\<Company>\<AppName>\<Version>\KeyName. The key names should be based on the same information you entered in the **Application Information** dialog box. For this example, type Software\Local Software, Inc.\Our New Release\4.0.950.0\Help Files in the **New Key** field.

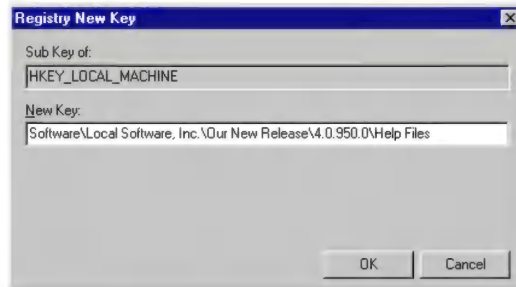


Figure 2.26. Adding a Key to the Registry.

After entering the information, click the **OK** push button to add the new key created to your application's project file.

11. The Registry Keys window updates to reflect the added key.

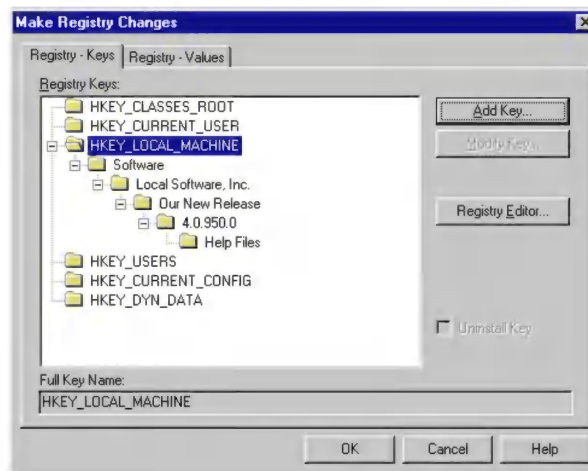


Figure 2.27. Displaying the New Registry Key.

Before we look at adding a value to the new key, let's set the uninstall key for your registry entries. Express makes it as easy as highlighting the key and checking the **Uninstall Key** box. In the Registry Keys window, highlight the **Help Files** key. When a non-root registry key is highlighted, the Modify Key... and Uninstall Key options are available. Select the **Uninstall Key** check box to add a Help Files uninstallation key. Your application's uninstall key should be under the general key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall. Then click the **OK** push button.

Note Be very cautious when selecting your uninstall key. If you select a key which has existing entries on the target system, such as the HKEY_LOCAL_MACHINE\SOFTWARE key, and the customer chooses to uninstall your application, all of the keys and values under that key, including those from other applications, will be deleted."

12. Once the key is created, you can set a value pair for it. In the Registry Keys window, highlight the key which will contain the new value. Click the **Registry - Values** tab to view the **Registry - Values** dialog box. The name of the key being modified is displayed in the Registry Key static text field, as shown in Figure 2.28.

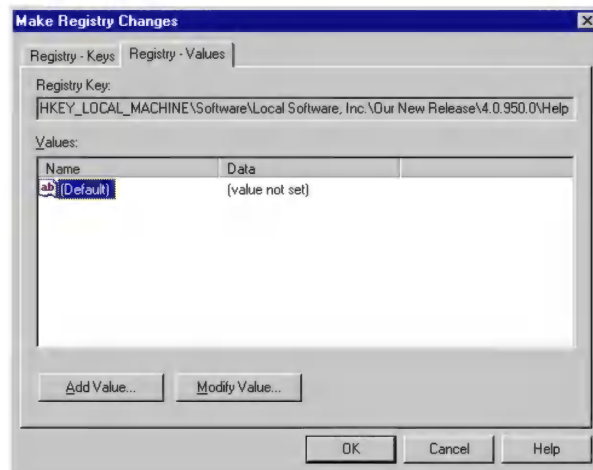


Figure 2.28. Adding a value pair to the registry.

Click the **Add Value...** push button to open the **Registry Value** dialog box. Select one of the radio buttons in the Value Type window to specify whether your data will be in string, binary, 32-bit decimal (DWORD Decimal), or 32-bit hexadecimal (DWORD Hex) format. For our example, select **String**.

13. In the **Value Name** field, enter the name of the value pair. For our example, enter `Help Version` in this field. In the **Value Data** field, enter the data to be linked to the value name. The entry for our example will be `1.0`. After making these entries, the **Registry Value** dialog box should look like Figure 2.29.

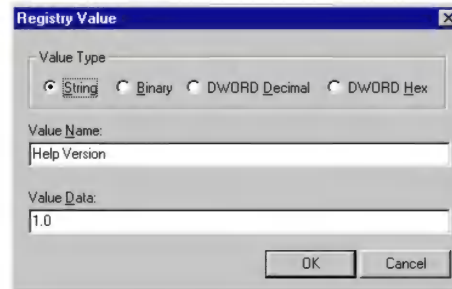


Figure 2.29. The Registry Value dialog.

After adding the value pair, click **OK** to apply the new information.

14. The Values window updates to display the new value pair added to the registry key. If no additional registry changes are needed, click the **OK** push button to close the **Make Registry Changes** dialog box.
15. Add a customized icon for this application. Express enables you to select the icons you want to place in your application's Windows 95 folder or Windows 3.1 group. Click the **General Settings** push button to reveal the **Specify Folders and Icons** dialog box. Since you specified WORDPAD.EXE in the Application Information dialog, Express will have automatically added an icon for this file.

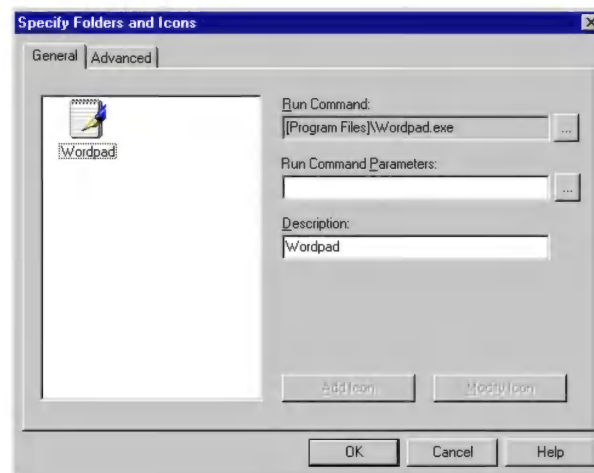


Figure 2.30. The General tab of the Specify Folders and Icons dialog box.

Let's add another icon for one of the text conversion files we are including with our application. Click the **Browse (...)** button to the right

of the **Run Command** field in the **General** tab of the **Specify Folders and Icons** dialog box to open the **Setup Files Browser** dialog box.

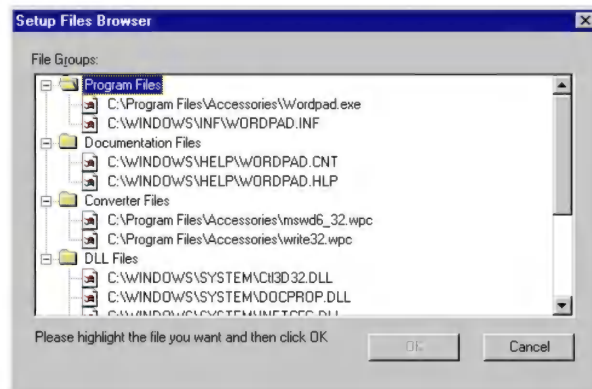


Figure 2.31. The Setup Files Browser dialog.

The files that you added to your groups in Exercise 3 are displayed in the File Groups window. Add an icon for the WordPad Help file by highlighting WORDPAD.HLP in the window. Click the **OK** button.

16. After you close the **Setup Files Browser**, the destination path of the selected file appears in the **Run Command** field of the General Settings dialog box. The **Description** field displays the name that will be assigned to the icon image. To add the WordPad help icon, click the **Add Icon...** push button. The dialog will update as shown in Figure 2.32.

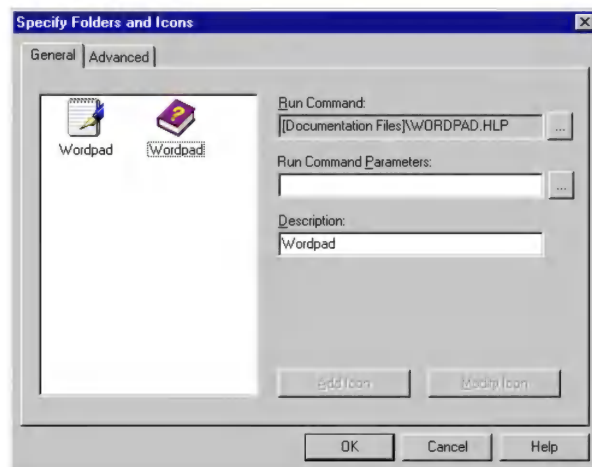


Figure 2.32. Displaying a New Icon.

If you want to choose an .ICO file to provide the icon image for your file, you can select the resource file on the **Advanced** tab. The Advanced icon settings dialog also enables you to specify an alternate working directory, select a shortcut key, place an icon directly the Start Programs menu (Windows 95 and Windows NT 4.0 only).

17. Create the disk images by running Disk Builder.
18. Test the installation by clicking the **Test Run** push button.
19. The final step of any installation created by Express is to create the distribution media. Click the **Copy to Floppy** push button in the Setup Checklist to open the following dialog box.

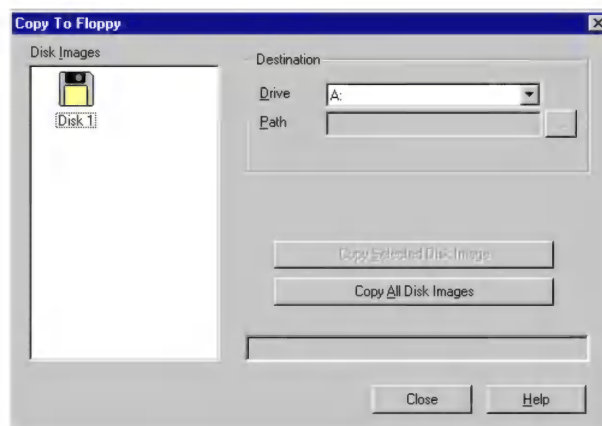


Figure 2.33. The Copy to Floppy dialog box.

20. The Destination window displays the drive and path that will receive your installation's setup files.

Note The entries in this field are dynamic and reflect the disk drives on your system. For example, if your system has two disk drives, A: and B:, both will be offered as options in the **Drive** field.

For example, if you select **A:** in the **Drive** field, Express will copy your setup directly to a disk in the A: drive. For our exercise, select **A:** as the destination drive.

Note Express enables you to create a self-extracting single file executable installation for your application. Selecting **Path for a 1 File Installation** from the **Drive** combo box displays a default path for the file and a password field. For more information on creating a self-extracting single file installation, refer to Chapter 3, "Building Your Installation."

21. The **Path** field enables you to select the location where you want to copy your setup. You are able to place the files directly onto a network or hard disk location. Either standard DOS paths or UNC paths can be used in this edit field.
22. After completing the fields in the Destination window, click the **Copy All Disk Images** push button or highlight an individual image in the Disk Images window and click the **Copy Selected Disk Image** push button. If you are copying all disk images to diskettes, Express will prompt you for each disk. Click the **Copy All Disk Images** push button to start the process.

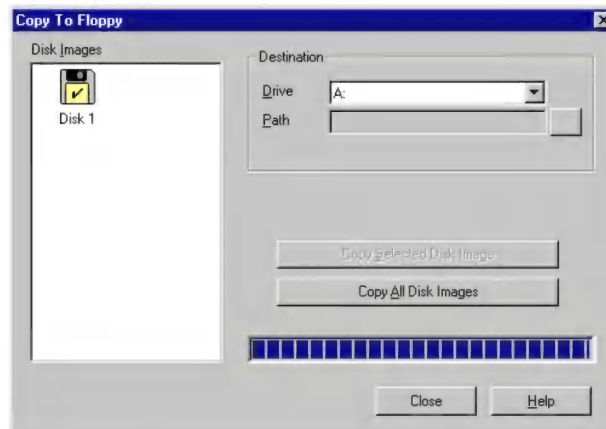


Figure 2.34. The Completed Copy to Floppy dialog box.

As each disk image is copied, a check mark is added to the particular icon in the Disk Images window. After the final image is copied, click the **Close** push button to finish the distribution disk creation process.

If you have any questions regarding the dialogs used in this chapter, please refer to Chapter 8, “Express Dialog Boxes” or the online help. You can access context sensitive help in any Express dialog is available by clicking the **Help** push button or the **F1** key.

Note If you would like to see an additional topic covered in this tutorial, or anywhere else in this manual, please contact the Express team at express@installshield.com. We are committed to providing exceptional documentation, and are always interested to find out what information YOU need.

While we strive to address all of the newest additions to our products in the printed materials, some features do not receive the amount of attention you might wish. The InstallShield website contains late-breaking information regarding all InstallShield products, including Express. To keep up to date on all Express news and upcoming additions, please check our website – URL:

<http://www.installshield.com/express> – on a regular basis.

Chapter 3

Building Your Installation

This chapter addresses some of the basic concepts you need to understand in order to create a professional installation using Express.

Experienced developers and Express users may already know much of the information contained in this chapter. For those who have not created your own installation before, or at least not with Express, the topics below cover some of the issues you may need to consider before designing your setup program.

Contents

Opening and Saving Your Setup Project File.....	3-2
What an Installation Does.....	3-8
How Express Helps You Create Your Installation	3-8
Groups, Components, and Setup Types.....	3-9
Target Directory vs. Source Directory	3-19
Express Directory Specifiers	3-20
Windows 95 Setup Guidelines.....	3-24
Creating a Single File Executable.....	3-27
Version Control	3-29

Opening and Saving Your Setup Project File

The foundation of your installation is called a setup project file. Your setup project file contains all of the information which you have entered into the Express dialogs. This section presents the basic information you will need to know to create any installation using Express. The first thing you will need to do in order to create your installation is open your project file.

To create a new project file, simply follow the steps below:

1. Launch Express. The following **Welcome** dialog will be displayed.

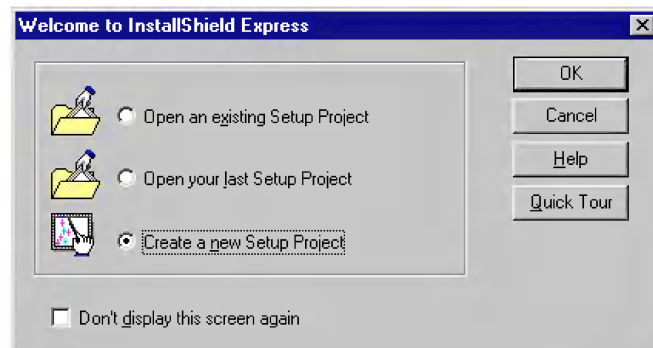


Figure 3.1. Welcome to InstallShield Express dialog.

Note the **Quick Tour** push button displayed in this dialog. Clicking this button launches the InstallShield Express Quick Tour. This tour provides an excellent look at Express's features and functionality and introduces the entire InstallShield family of products. In fact, this tour was created using DemoShield.



Figure 3.2. Sample Screen from the InstallShield Express Tour.

DemoShield is the complete package for presenting and promoting your applications. DemoShield features fully interactive point-and-click controls, screen capture capabilities, visual design environment and comprehensive sample demos. The same ease of use that you are experiencing with Express is built into DemoShield. To find out how you can use DemoShield to show and sell your applications, visit the DemoShield website at <http://www.demoshield.com> or call 800-250-2191 for more information.

Please take a few minutes and view the Quick Tour. It gives an excellent overview of Express. If you elect not to take the tour at this point, you can select it later by choosing **Quick Tour** from the **Help** menu.

2. Since you are creating a new installation, select **Create a new Setup Project**, then click the **OK** button.

Note If you do not want this dialog box to be displayed when you open Express, select the **Don't display this screen again** checkbox before you select OK. Selecting the checkbox causes this dialog box to be permanently hidden from this point forward. If you select this option, all of the functions appearing in this dialog box are always available in the Express menu bar.

- Express will then launch the **New Project** dialog box.

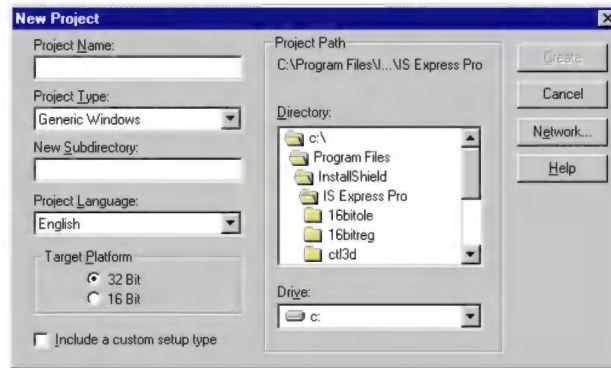


Figure 3.3. The New Project dialog box.

- Select the drive and directory where you want to store this project using the standard Directory and Drive controls. The current path for the project file is displayed at the top of the box.
- If you want to create a new subdirectory for your project file, simply enter its name in the **New Subdirectory** field. Express will automatically create the directory under the path you specified in step 4 and place your project file in it.
- Enter the name of your project in the **Project Name** field. Express will use this name, followed by the .IWZ extension, as the filename for this project.
- Select the desired development environment from the **Project Type** drop-down menu. This selection will determine which set of InstallShield Objects will be included with your setup project. If you are not using any of the platforms supported by Express, you can select **Generic Windows**.
- In the **Project Language** field, you can specify the language you want your installation to be created in. The dialogs displayed to your customers during the installation process can be in English, French, or German. More languages will be added in the near future.
- Select the **16 Bit** or **32 Bit** radio button to specify the target platform for your setup.

Note If the development environment selected in the **Project Type** field does not support both 16-bit and 32-bit settings, these radio buttons will not appear in the **New Project** dialog box.

If you are uncertain as to which setting you should select for your project, consult the following table.

If your application is...	and your application will be installed on...	select this Target Platform setting.
16-bit	Windows 3.1 or Windows 3.11	16 Bit
16-bit	Windows 95 and higher or Windows NT 3.51 and higher	16 Bit
32-bit	Windows 95 and higher or Windows NT 3.51 and higher	32 Bit

Note We recommend that you create both a 16-bit installation and a 32-bit installation if you plan to have your application installed on both 16-bit and 32-bit platforms. However, if you are only going to create a one installation to cover both target platforms in this situation, we suggest using the 16-bit target platform setting. Note that you will not have access to 32-bit InstallShield Objects if you select the 16-bit setting.

- When you are satisfied with your entries, click the **Create** push button. The **New Project** dialog box closes and the Setup Checklist is displayed.

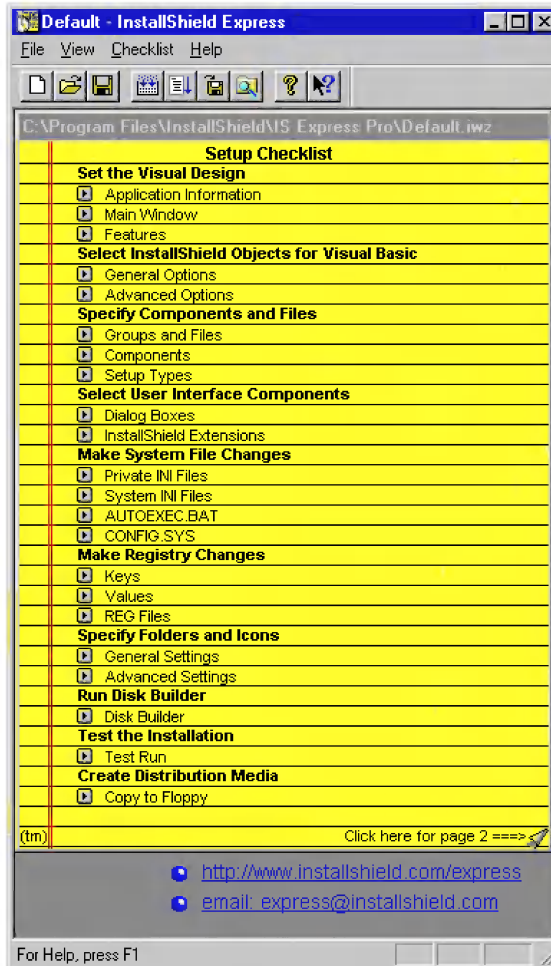



Figure 3.4. The Setup Checklist.


You will notice that the file location information entered in the **New Project** dialog box appears in both the title bar of the window and the top line of the Setup Checklist.

Opening an Existing Project

Once you have created a project file, you can re-open it using any of the following methods:

- If the project you want to open is one of the last four you have saved in Express, you can select the project name from the most recent file list displayed at the bottom of the **F**ile drop-down menu.
- Clicking the  button and finding your file using the standard Open dialog.
- Selecting **O**pen... from the **F**ile menu and finding your file using the Open dialog.
- Select the project name from the **D**ocuments list in the Windows 95 start menu. This will launch Express and open the selected file.


Saving a Project

To save an open project, either click the  toolbar button or select **S**ave from the **F**ile menu.

Closing the Current Project

You can close your project file by select **C**lose from the **F**ile menu. You can then create a new project file, open a different existing file, or exit Express.

To close Express:

- Select **E**xit from the **F**ile menu.
- Click on the **X** button at the upper right hand corner of the Express main window.
- Click on the  icon at the upper left hand corner of the Express main window and select **C**lose from the drop-down menu.

What an Installation Does

An effective installation program performs the following four tasks upon its execution:

- Checks customer's disk for sufficient space available and determines the software configurations present before copying any files.
- Updates an existing installed application.
- Copies files to the specified directories on the customer's disk.
- Modifies existing files and makes the appropriate entries to the registry.

Failure to address any of these issues can cause serious problem when attempting to install any software. Not only can your software's performance be adversely effected by a poor installation process, but your reputation itself can suffer if your customers cannot easily install your product. Express ensures that you quickly create a very effective installation.

How Express Helps You Create Your Installation

InstallShield Express combines the ease of use you have come to expect from well designed Windows applications with the best installation technology available. Express takes the complicated process of scripting an installation and breaks it down into very user-friendly and manageable tasks. Using Express, you specify application files by dragging and dropping, and create most or all of your setup program by pointing and clicking or responding to simple dialog boxes.

In fact, in many of the Express dialogs, you may not have to make any changes at all. All default settings in InstallShield Express are set to conform to the Windows 95 Application Setup Guidelines for Independent Software Vendors. For more details, refer to the Windows 95 Setup Guidelines section in this chapter.

Express also makes you look good by giving your application's installation the Windows 95 look and feel. All of the dialog boxes displayed during an Express setup contain the industry standard controls familiar to millions of Windows users. This creates customer confidence in your application.

You can add custom bitmaps and billboards to your setup to advertise, inform, entertain, or visually integrate the installation with your application.

There's no need for anyone to know how easy it was to create. All your customer will see is a professional setup that installs perfectly onto his system.

Groups, Components, and Setup Types

Even though Express is designed to be the most convenient installation development system available, your setup will require some preparation. A bit of effective planning before you begin creating your installation can greatly smooth the entire process.

You will need to carefully consider the structure of your application files to be added to the customer's system:

- Where do your application files need to be installed on the target system?
- What support and redistributable files are required by your application?
- Do you want to offer your customer the option of selecting among multiple setup types?

In InstallShield Express, groups, components, and setup types provide the framework for copying your files. You will specify the files and their destination location(s) by placing them into groups. To create multiple setup types, you add these groups to components, which represent logical elements of your application. Using components, you can offer multiple setup types to your customer.

Figure 3.5 graphically represents the relationship between these concepts:

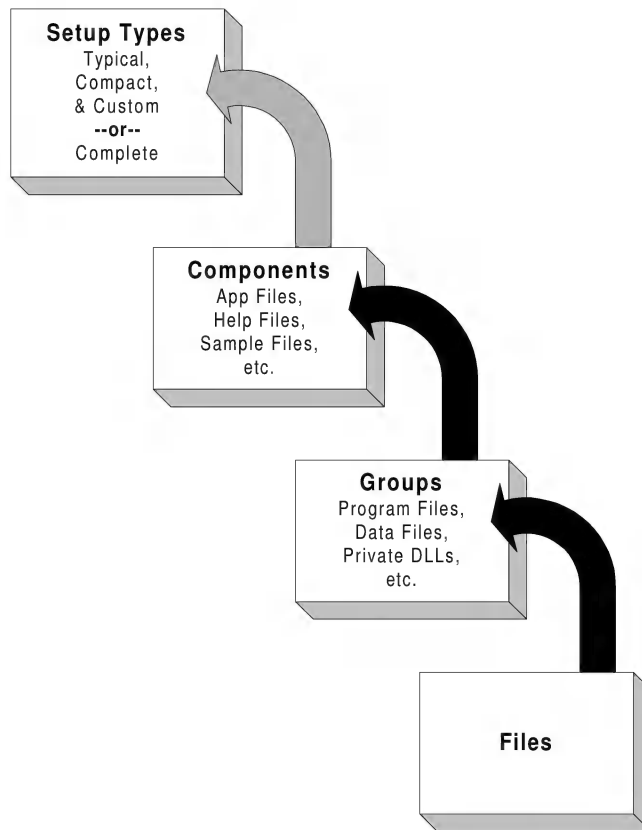


Figure 3.5. Setup Types, Components, Groups, and Individual Files.

Setup Types

You have probably already encountered multiple setup types while installing software. Setup types allow the customer to select from among several installation "packages." A dialog box appears on the customer's screen during the installation prompting them to select their preferred setup type. Figure 3.6 is an example of the different setup types presented in Microsoft Access.

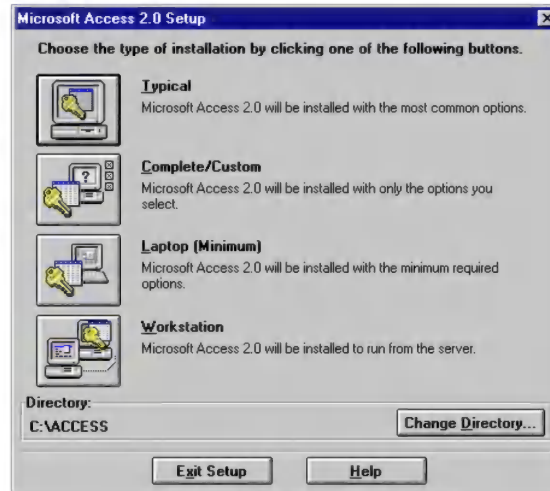


Figure 3.6. A Setup Type Example.

You need to consider whether you want to include a custom setup option. If there are any files which are not needed to run the application, such as samples or templates, you might consider placing them in separate components, which the customer could elect not to install.

If you choose to include multiple setup types, Express supports these three:

- **Typical:**
Usually a complete setup, including all components.
- **Compact:**
Usually consists only of those components necessary to run the application.
- **Custom**
Allows the customer to select which components will be installed.

While Express enables you to create different setup types, it is up to you to determine what each setup type actually installs. You control this by grouping your files into components. The following illustration represents the relationship between individual files, setup groups, setup components, and setup types.

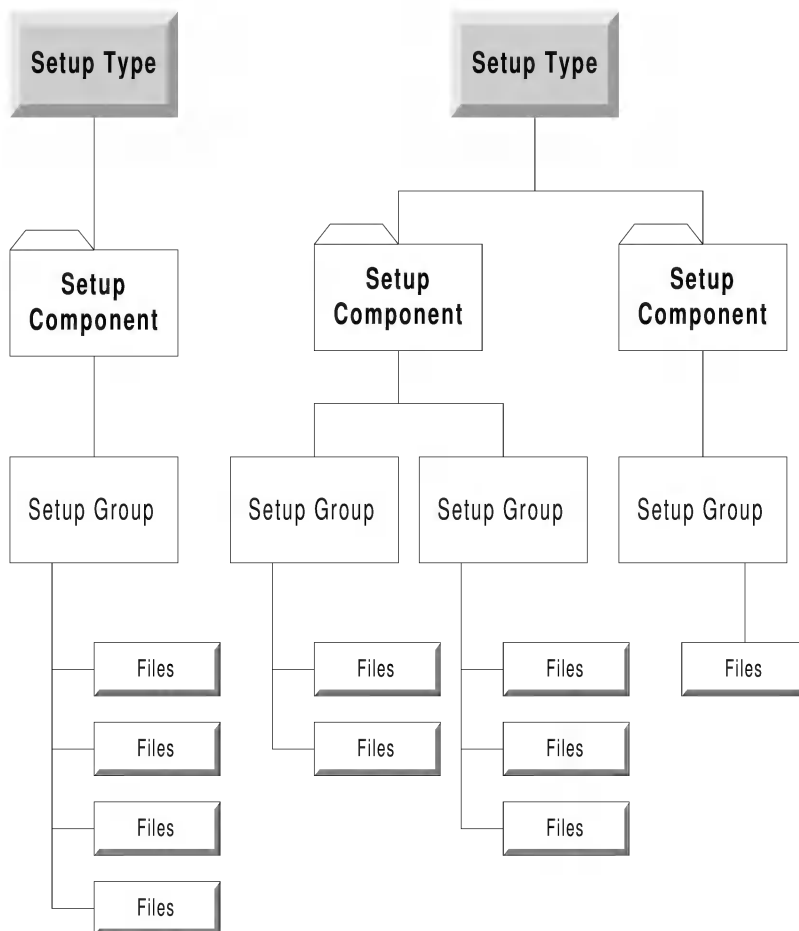


Figure 3.7. Organizing Your Application's Files.

What if I only need a single setup type?

You do not have to include multiple setup types in your installation. Even if you chose the option to offer them when you first begin your project, you can modify it to display only one setup type (Complete) at a later point.

If you are not offering multiple setup types, you won't need to worry at all about components and setup types. Express will automatically add all of your file groups to a single component, which is included in the Complete setup type.

If this is the case, all you will need to consider while creating groups for your application files is the destination directories of your files. Simply place all files which you want to install into a particular directory into the same group. Create an additional group for each different destination directory you are using, and add all the files targeted for each directory to the respective groups. Express takes care of the rest!

Components

Components are the building blocks of setup types. Components are groups of related files that represent different parts of the application, such as Help and Readme files, templates and examples, tutorials, and the main application itself.

If you are not offering multiple setup types, you don't even need to worry about components. All file groups are contained in a single component, which is automatically included in the Complete setup type.

If you are offering multiple setup types, you will specify which components are included with each setup type in the Setup Types dialog.

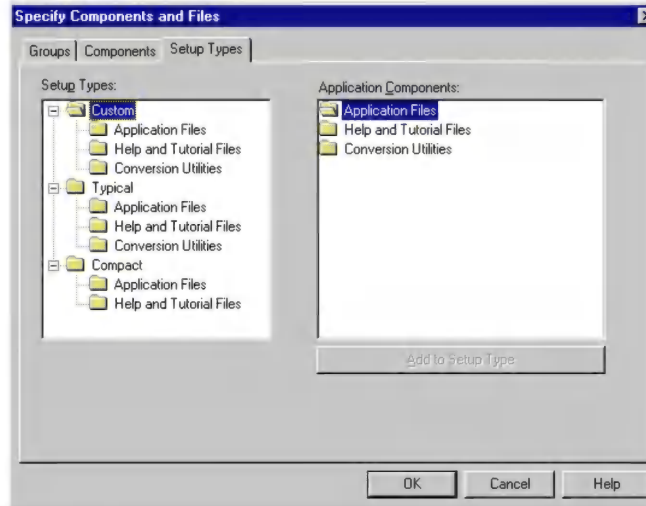


Figure 3.8. Sample Setup Types dialog box.

You create your components by adding groups to each component in the Components dialog.

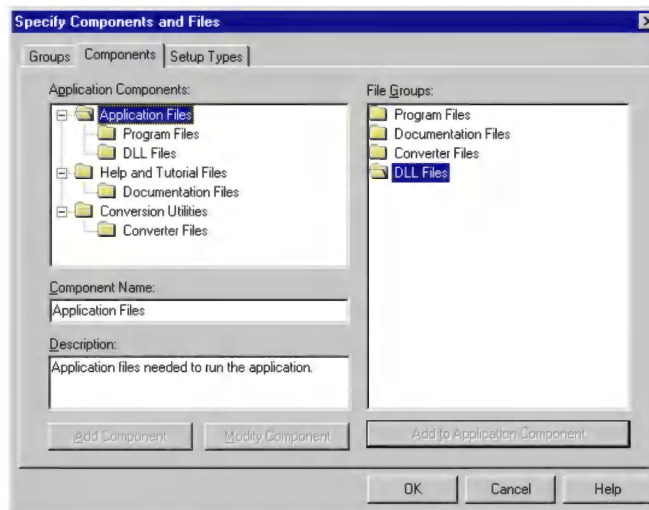


Figure 3.9. Sample Setup Components dialog box.

Groups

Groups are similar to components, in that they are collections of related files. However, in Express, files and destination directories are assigned directly to groups, then the groups are assigned to components.

All of the files in a group will be copied to the same directory. Copying your files into the group will be quick and easy if you already have them in separate directories which represent your groups.

Basically, groups allow you increased flexibility. By specifying different destination directories, you can create components which include files to be placed in more than one directory, rather than having to specify one directory for the entire component.

You can also choose to allow the customer to select which individual groups to install in a custom setup by selecting the check box on the Custom Setup user dialog settings tab. This gives you the choice to designate certain groups of files as optional.

Planning your files groups and components becomes even more critical when you are offering setup type options.

Sample Group and Component Plan

Notes

- This sample may be much more complex than your application structure. The number and configuration of groups and components were chosen in order to illustrate several concepts in one example. Don't be fooled into thinking you have to make your setup unnecessarily complex. The simpler your installation structure is, the easier it is to plan.
- You will notice file directories in this sample with names like <INSTALLDIR> and <WINSYSDIR>. These are called Express directory specifiers. For the purposes of this section, just understand that these specifiers represent directories on the customer's system. For a complete explanation of this topic, refer to the Express Directory Specifiers section later in this chapter.

Let's assume you have an application called MyApp, and you have decided (after some gentle urging from your supervisor) to use Custom and Compact setup types.

You initially divide your application into five components:

- Help Documents
- Examples
- Data Files
- Utility Programs
- Application Files

All five components will be included in both the Typical and Custom setup types. All except for Examples will be included in the Compact setup type.

You plan to install all of the files from the Utility Programs and Application Files components in the <INSTALLDIR> directory. You wish to place Examples in <INSTALLDIR>\EXAMPLES, and Data Files in <INSTALLDIR>\DATA. Since the files in each of these components all go in the same directory, you have created and assigned only one group for each.

You have been instructed to place each piece of the documentation in its own directory, so you have created three groups for the Help Documents components:

- Readme Files (to be placed in <INSTALLDIR>\README)
- Help Files (<INSTALLDIR>\HELP)
- Tutorial (<INSTALLDIR>\HELP\TUTORIAL).

Thus, your initial plan could be represented by the following chart in Figure 3.10.

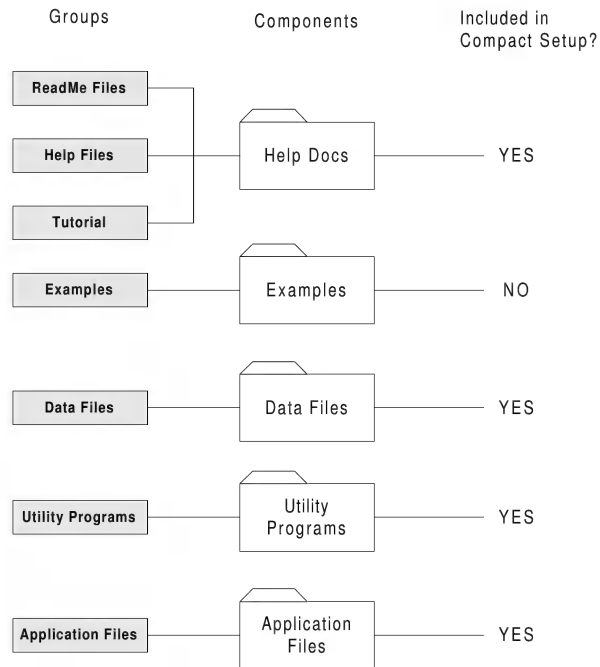


Figure 3.10. The Initial Setup Plan.

As you are preparing and testing your setup, you realize that some of the data files are shared .DLLs which really should be installed in the <WINSYSDIR> directory. So you delete the Data Files group and create two new groups, Text Files and DLLs. You specify <INSTALLDIR>\DATA for the Text Files group and <WINSYSDIR> for the DLLs group, and assign both to the Data Files component.

Your supervisor then informs you that the extensive tutorial which he demanded of you is not an absolutely necessary piece of documentation, so it should not be included in the Compact setup type. Rather than create a whole new component, you decide to delete the Tutorial group from Help Documents and add it to the Examples component, which you rename Support Files.

Shortly thereafter, Ted from marketing complains that his beta version of your product isn't working properly. When you go to look at his system, you realize that he chose a custom setup and did not install the Utility Programs component. You decide that, in order to try to prevent anyone else from making this mistake, you will delete the Utility Programs component and add the Utility Programs group to the Application Files component.

Ted then informs you that you have to include a marketing survey, and your supervisor backs him up, although he does not insist that the survey be required for installation. So you add the Survey group and component and assign the component to the Typical and Custom, but not Compact, setup types.

Now your updated plan chart looks like the following:

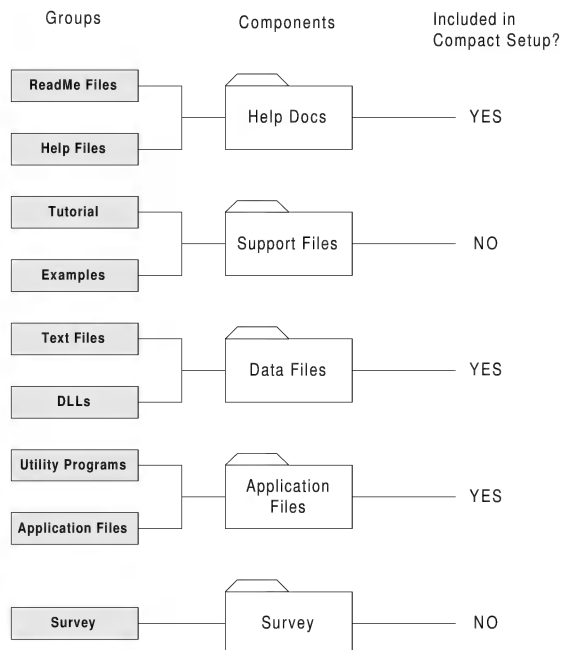


Figure 3.11. Revised Setup Plan.

Fortunately, InstallShield Express enables you to make all of these changes without having to rewrite sections of code. A few clicks in the appropriate dialog boxes, perhaps a couple of drags and drops, and you're ready to go again.

When you are testing your setup, experiment by selecting different components for a custom setup. Make sure that all selected files are being copied (and being copied to the right directory).

Although Express will allow up to 100 components, and up to 100 groups per component, try to use only as many groups and components as you need to accomplish what you want.

Target Directory vs. Source Directory

When you begin using Express, you may notice that some of the files you specify in various dialogs, such as Groups and Files, are displayed with their current path:

C:\Program Files\ MyApp\ Samples\ Sample1.ext

In other locations, such as the **Application Executable** field of the **Application Information** dialog box, the file you select is displayed in a different format:

[Program Files]\ MYAPP.EXE

Express uses two different filename formats because the installation process requires knowledge of where certain files are in the source directory (i.e. on your system), and where other files will be placed in the target directory (on your customer's system).

Target directory

In most setups, the customer selects directory in which your product will be installed. While a default setting is provided, unless you de-select the **Choose Destination Location** user dialog, your customer has the option to install your software anywhere he has write privileges.

The installation you create using InstallShield Express provides your customer a chance to change the target directory from the default setting before any files are copied. The target directory is selected based on the information your customer enters in the **Choose Destination Location** dialog that appears during the setup process.

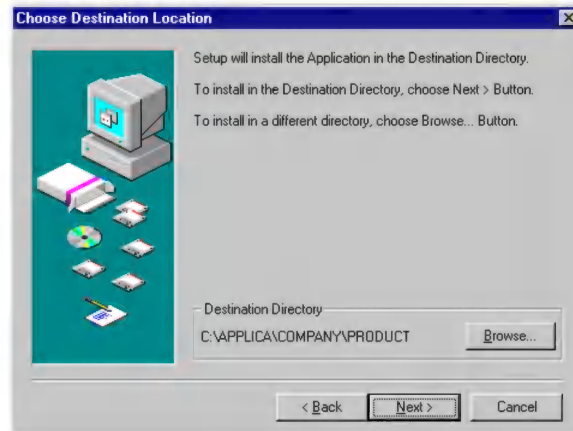


Figure 3.12. Enabling the User to Select the Target Directory.

In order to allow you to use information about the target system in your setup, InstallShield Express utilizes directory specifiers. The Express directory specifiers, such as <INSTALLDIR>, are replaced by target system locations during the installation.

You can use the Express directory specifiers in many Express dialogs – anyplace that you need to indicate a system-dependent target location. These specifiers are covered in greater detail in the following section.

Source directory

The source directory refers to the directory from which the files are being copied to the installation disks. Basically, the source directory of a given file is its current location on your system

Express Directory Specifiers

When installing your application, you may need to write the location of a particular file or directory on the target system to the registry, an .INI file, or a source file. Additionally, many Express functions require knowledge of the target location of files. In order to accommodate these needs, Express employs directory specifiers.

Since so many customers today have multiple hard drives, disk drives, and network drives on their systems, you have no way of knowing where a particular customer wants to install your application, nor even where the main Windows and System directories are on that system.

The Express directory specifiers act as placeholders during the creation of the installation setup. Once the actual installation of the application on the customer's system begins, these directory specifiers are replaced by information about the customer's system. They can be used in any Express dialog where you need to express the location of a file after it has been copied to the target system.

The following table displays the set of InstallShield Express' directory specifiers and their functions:

Specifier	Function
<INSTALLDIR>	Replaced with the main installation directory selected by the customer during the setup process.
<WINDIR>	Replaced with the main Windows directory on the target system (e.g. C:\WINDOWS).
<WINSYSDIR>	Replaced with the SYSTEM directory on the target system (e.g. C:\WINDOWS\SYSTEM). If the target system is running Windows NT, this specifier will indicate the SYSTEM32 directory if you have a 32-bit setup, and SYSTEM directory if you have a 16-bit setup. If you need to include 16-bit .DLLs with your 32-bit application, use <WINSYS16DIR> instead.
<WINDISK>	Replaced with the drive letter of the disk containing the WINDOWS directory (e.g. C:).
<WINSYSDISK>	Replaced with the drive letter of the disk containing the SYSTEM directory (e.g. C:).
<WINSYS16DIR>	On a Windows NT target system, this specifier is replaced with the 16-bit SYSTEM directory. On a Windows 95 or Windows 3.1 system, this specifier is exactly like <WINSYSDIR>.

Specifier	Function
<ProgramFilesDir>	Replaced with the PROGRAM FILES directory on the target system (e.g. C:\ PROGRAM FILES). In Windows 95, the Program Files directory location is stored in the registry under the "ProgramFilesDir" value name. In Windows NT, Express appends the PROGRAM FILES subdirectory to <WINDISK>, and in Windows 3.1, Express appends ProgramF to <WINDISK>.
<CommonFilesDir>	Replaced with the COMMON FILES directory on the target system (e.g. C:\ PROGRAM FILES\ COMMON FILES). In Windows 95, the Common Files directory location is stored in the registry under the "CommonFilesDir" value name. In Windows NT, Express appends the Common Files subdirectory to <WINDISK>\ Program Files, and in Windows 3.1, Express appends CommonF to <WINDISK>\ ProgramF.
<SRCDIR>	Replaced with the source directory from which the application files are taken. For example, if the source files are to be installed from diskettes located in the customer's A: drive, the "SRCDIR" value will be A.
<SUPPORTDIR>	This directory specifier is used in conjunction with storing temporary installation files in _SETUP.LIB. For example, if you have a custom .DLL (named MYDLL.DLL) that is to be used during the installation and then deleted, the extension filename for this .DLL would be <SUPPORTDIR>\ MYDLL.DLL. For more information on using extensibility, refer to Chapter 5, "InstallShield Extensions."
[group name]	A bracketed group name (e.g. [Program Files]) can also be used as a specifier. This specifier is replaced by the directory into which the particular group is copied.

You can indicate a subdirectory of any of these specifiers by modifying them as you would any standard path. To specify the HELP subdirectory of your main directory, you would enter <INSTALLDIR>\ HELP. Or, to specify the TEMPLATES subdirectory under the target location of the Support Files group, you would enter [Support Files]\ TEMPLATES.

You can use the Express directory specifiers in any of the following locations, to name just a few:

- The **Destination Directory** field of the **Select Groups and Files** dialog box.
- Any of the three edit fields of the **Setup Complete** user dialog box settings tab.
- The **Value** edit field of the **Private .INI Files** dialog box.
- The **Value** edit field of the **System .INI Files** dialog box.
- The **Name=Value** or **Command** edit fields in the **AUTOEXEC.BAT** dialog box.
- The **Setting** edit field of the **CONFIG.SYS** dialog box.
- The **Value Data** edit field of the sub-dialog boxes for adding or modifying a registry key value in the **Registry - Values** dialog box. Refer to Chapter 6, “The Registry” for more information on using directory specifiers in the registry.
- The **Run Command Parameters** edit field in the **General Icon Settings** dialog box.
- The **Working Directory** edit field in the **Advanced Icon Settings** dialog box.

Additionally, Express automatically uses some of these specifiers in certain locations, such as the **Application Executable** field in the **Application Information** dialog box and the **Run Command** field in the **General Icon Settings** dialog box.

Creating a Windows 95 Compatible Application

In order to get a Windows 95 logo for your application, you need to follow the Microsoft Windows 95 Application Setup Guidelines for Independent Software Vendors. For more information regarding these guidelines or how to have your application tested for the Windows 95 logo, please visit the Veritest website at <http://www.veritest.com> or email them at logolab@veritest.com. You can also contact Microsoft by visiting their website at <http://www.microsoft.com/windows> or mailing them at winlogo@microsoft.com

Fortunately, Express makes qualifying for the Windows 95 logo easy for you by providing default selections which conform to the Windows 95 guidelines. Express also makes automatic registry entries for you, including the App Paths, uninstallation, and user information values.

The following list is a quick look some important points of the guidelines. It is **not** a complete or comprehensive listing.

Note None of the following items are required by Express to create an installation, they are merely suggested if you want to comply with Microsoft's guidelines.

- **Keep the number of files copied to the WINDOWS or SYSTEM directories to a minimum.**

As a general rule, do not copy files to the WINDOWS or SYSTEM directories. You may still place shared .DLLs in the SYSTEM directory. Express compares the version number of any files placed in the SYSTEM directory, and only installs your version if it is more recent than the version currently on the target system.

- **Do not make changes to the WIN.INI, AUTOEXEC.BAT, or CONFIG.SYS files.**

In Windows 95, you do not need to modify the path in AUTOEXEC.BAT. Instead, use the per-application paths (App Paths) key in the registry. Express automatically enters the path for your primary executable under the App Paths key. You can also specify additional paths under this key.

Windows 95 also offers you the ability to dynamically load your device drivers without modifying the CONFIG.SYS file. Design your application to load its drivers when the application is launched.

Information which would have been entered into the WIN.INI file in Windows 3.1 or NT should now be loaded into the registry or your private .INI files.

- **Keep the number of icons in your folder to a minimum.**

Usability studies have shown that novice users are confused by lots of icons in a program folder.

Ideally, you would have just one icon added directly to the Start Programs menu (you can do this through the Advanced icon settings dialog) and allow access to tutorials, help files, and other applets through the help or tools menu. This will ensure that the customer will not have to go through an additional step, and will always know how to launch the application.

- **Create all of the necessary directories in the path specified by the customer.**

Not all customers have what we might consider a typical directory structure. For that reason, they might decide to install your application to a directory other than the default suggestion provided by your setup. If a different directory is selected, your installation must create all of the required subdirectories in the new path. While that sounds like a very difficult task, Express handles it quite easily through the use of Express directory specifiers. These are explained in the previous section of this chapter.

- **Follow the new filename, name space, and file extension guidelines.**

Windows 95 and Windows NT both support the use of long filenames. Express also supports these guidelines for 32-bit installations.

Note When working with 16-bit installations, all directories and filenames you add must conform to the 8.3 naming convention. Failure to do so will cause an error in the building of your 16-bit installation.

- **Do not remove the automatic uninstallation option.**

Express comes complete with unInstallShield. This program satisfies the Windows 95 Setup Guidelines for uninstallation capability.

The best part is that you don't have to do anything – just leave the check box on the Features dialog in its default (selected) position.

- **Name your installation program SETUP.EXE.**

Naming your installation file SETUP.EXE makes it much easier for the Windows 95 Add/Remove Programs applet to locate your installation without relying on the customer to enter the correct file. The applet scans for executable files named SETUP when attempting to automatically find the correct installation file.

- **Store private information in the APPLICATION directory.**

The APPLICATION directory should be located under the PROGRAM FILES directory. This makes the process of uninstalling the software as well as installing future upgrades much easier.

- **Always supply defaults.**

In case you are new to the Windows 95 interface, you may not have noticed that a default setting always appears in any selection box. This is by design. By always supplying a default setting, you have a chance to “suggest” the best option available. This way, you can present you customer with the optimal settings for your application.

- **Do not assume the program diskettes are in Drive A.**

With the advent of network installations, CD-ROM, and easily customizable hardware configurations, you can no longer assume that your application will be loaded from the customer's A: drive on a given system. Express enables you to create an setup that allows the customer to specify from where the program is to be installed.

- **Keep the customer informed during the installation.**

Notify the customer as to how much space the installation will take and provide a progress indicator. Express provides these services to your customer, as well as enabling you to add your own customized billboards to display during the file transfer process.

Creating a Single File Executable

InstallShield Express features the ability to easily create a single-file, self-extracting executable file for your application. If you have ever needed to post your application on a FTP site or distribute your application across a network, the ability to create a single self-extracting executable file by simply selecting an option from a dialog box will be of great interest to you. In addition, Express supports password-protection for the single file executable it creates.

To create a single-file, self-extracting executable, perform the following steps.

1. Select the **Copy to Floppy** push button, which is located in the Create the Distribution Media section in the Setup Checklist.

Note Before you can launch the **Copy to Floppy** dialog box, you must have first created the disk images for your installation in Disk Builder.

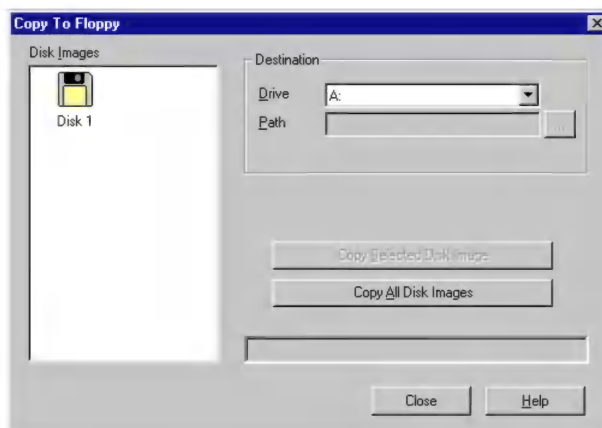


Figure 3.13. The Copy to Floppy dialog box.

The icons displayed in the Disk Images window are those that were built by Disk Builder according to your specifications.

2. Select **Path for a 1 File Installation** from the drop-down menu available in the **Drive** combo box. The Destination Window will change to reflect this selection.

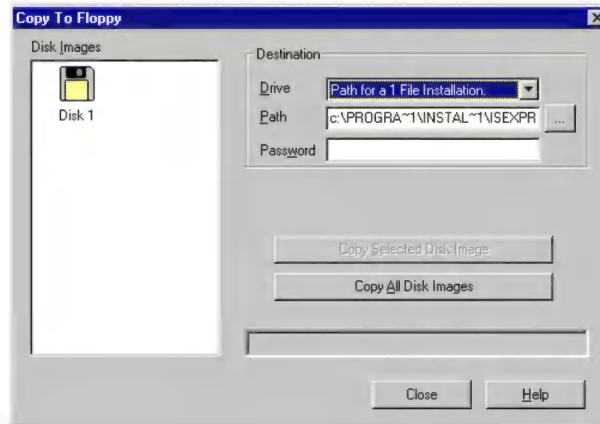


Figure 3.14. Single-file executable destination path and password.

A default path for the file and a password field appear on the tab when this option is selected.

3. The **Path** edit field enables to enter the path where you want to copy your setup, allowing you to place the files directly onto a network or hard disk location. You can use standard DOS paths or UNC paths in the edit field. The browse (...) button to the right of the field opens a **Choose Directory** dialog box, which allows you to select the path without typing.
4. If you would like to add a password to the installation file for security, enter it in the **Password** field.
5. After entering the information in the destination fields, click the **Copy All Disk Images** push button to start the disk creation process. When the process is completed, a dialog box will be displayed.

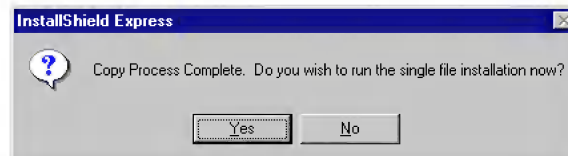


Figure 3.15. Copy process completed dialog box.

6. If you select **Yes** from the dialog box, the self-extracting .EXE will be launched on your system. Selecting **No** will close the dialog box and return you to the **Copy to Floppy** dialog box.
7. Click the **Close** push button to return to the Setup Checklist.

That is how easy it is to build your application's self-extracting executable file. For more information on creating your distribution disks, refer to Chapter 2, "A Quick Tutorial."

Version Control

InstallShield Express includes version control only for those files copied to the Windows System <WINSYSDIR> directory.

Express will check for an existing version of each file copied to this directory. It will replace an existing file only if you have included a more recent version with your installation. Of course, it will copy the file normally if there is no existing file of the same name.

Chapter 4

InstallShield Objects

Many applications created in visual or object-oriented development environments use the functionality available in additional DLLs or other files. Often, these resources and controls require registry entries or system-file configuration in order to provide this functionality to your application.

Determining the necessary dependent files and system modifications for your application can be the most challenging part of creating a setup. InstallShield Express provides a simple method for including these dependencies: InstallShield Objects. This chapter describes InstallShield Objects and explains how to add them to your installation.

Contents

What is an InstallShield Object?.....	4-2
Borland C++ 5.0 Objects	4-4
Delphi 2.0 Objects	4-7
Paradox 7.0 Objects	4-12
Visual Basic 4.0 Objects.....	4-15
Visual C++ 4.1 Objects.....	4-19
Advanced Settings.....	4-21
BDE/IDAPI Settings	4-22
SQL Links.....	4-27
Selecting Individual ODBC Drivers.....	4-28

What is an InstallShield Object?

InstallShield Objects encapsulate all of the knowledge necessary to deploy development environment-specific components. They present this capability to you in a simple, point-and-click dialog.

When you created your project, you selected a Project Type, which corresponded to a particular development environment. The specific objects available for use in your setup are based on the selection you made.

Note Express automatically registers all OCX controls containing the “OLESelfRegister” string in the version resource. This automatic handling of self-registering files applies for **all** development environments. Refer to Self-registration of .DLLs and .OCXs in Chapter 7, “Frequently Asked Questions” for more details.

When you open the **General Options** dialog, Express displays the objects available for your specified environment. Express Professional version 1.0 includes the following objects:

- **Borland C++ 5.0**
(32-bit) BDE/IDAPI, SQL Links, RTL, OWL, ClassLib, VBX, VBDT, and more.
(16-bit) BDE/IDAPI, RTL, OWL, ClassLib, VBX, VBDT, and more.
- **Delphi 2.0**
(32-bit) Full or partial Borland Database Engine (BDE) installations, SQL Links, ReportSmith Runtime Report Viewer, and ODBC.
(16-bit) Borland Database Engine (BDE) installations and ODBC.
- **Paradox 7.0**
BDE/IDAPI, SQL Links, Paradox Runtime, and ODBC.
- **Visual Basic 4.0**
Data Access Objects (DAO/J et), ODBC, and most popular controls, plus automatic Visual Basic project file review.
- **Visual C++ 4.1**
(32-bit) DAO/J et, MFC 4.0, ODBC, and other popular controls.
(16-bit) MFC and ODBC.

You can incorporate the functionality of most of these components in your setup by merely selecting the appropriate check box in the General Options dialog. When you select an InstallShield Object, Express automatically adds the necessary files, groups, and registry or system-file modifications.

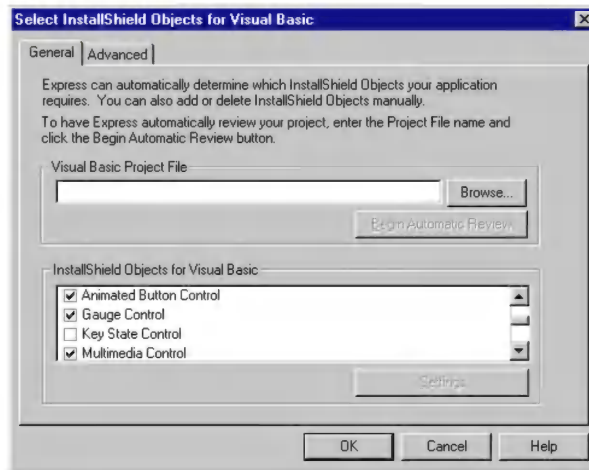


Figure 4. 1. Selecting InstallShield Objects.

To view details (date, size, and version, if applicable) of the files added by the object(s) you have selected, click the **Advanced Options** tab.

Note The Advanced InstallShield Objects dialog is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. If you need to delete files from any of the automatically generated groups, you can do so in the Groups and Files dialog.

Some InstallShield Objects require you to specify settings, such as the drivers you wish to include with your installation. The following sections in this chapter contain additional information about these settings, organized by development environment.

Borland C++ 5.0 Objects

InstallShield Objects for Borland C++ 5.0 include support for the Borland Database Engine, SQL Links, OWL, VBDT, and other components.

Note If your application does not require any of these add-ons, do not select any of the InstallShield Objects. They will add unnecessary files to your setup.



When you select a Borland C++ object, Express adds the necessary files and registry/system-file changes to your setup for the object selected. With the exception of the BDE/IDAPI, SQL Links, and ODBC objects, the InstallShield Objects for Borland C++ require no settings.

Selecting Borland C++ Objects

To select the objects to be included with your setup, click the **General Options** push button in the Setup Checklist to open the following dialog:

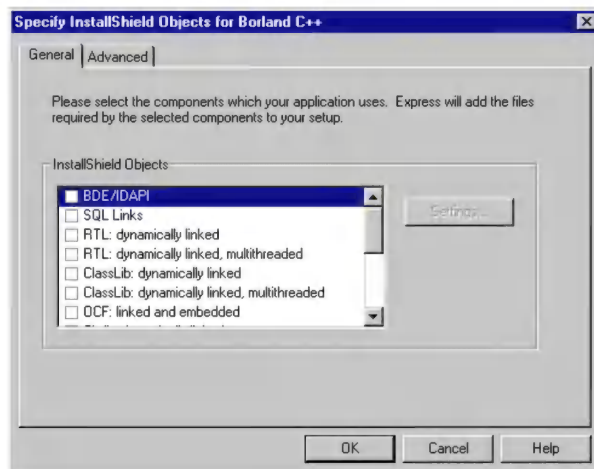


Figure 4.2. Borland C++ Objects dialog box.

In the **Specify InstallShield Objects for Borland C++** dialog box, select the controls that your application requires by clicking the check boxes for as many of the following as apply:

- **BDE/IDAPI**
Choose if your application uses the Borland Database Engine. Selecting this check box launches the BDE Settings dialog boxes. This process is covered in step-by-step detail in the BDE/IDAPI Settings section later in this chapter.
- **SQL Links***
Choose if your application requires SQL links to various database servers. Selecting this check box launches the SQL Links Settings dialog. This process is covered in detail in the SQL Links Settings section later in this chapter.
- **RTL: dynamically linked**
Choose if your application dynamically calls the run-time libraries at run-time.
- **RTL: dynamically linked, multithreaded**
Choose if your application dynamically calls the multithreaded version of the run-time libraries at run-time.
- **ClassLib: dynamically linked**
Choose if your application dynamically calls the class libraries at run-time.
- **ClassLib: dynamically linked, multithreaded**
Choose if your application dynamically calls the multithreaded version of the class libraries at run-time.
- **OCF: linked and embedded**
Choose if your application links and embeds the ObjectComponents libraries at run-time.
- **OWL: dynamically linked**
Choose if your application dynamically calls the ObjectWindows libraries at run-time.
- **OWL: dynamically linked, multithreaded**
Choose if your application dynamically calls the multithreaded version of the ObjectWindows libraries at run-time.

* This object is only functional if BDE/IDAPI is selected.

- **Borland Windows custom controls**
Choose if your application uses the Borland Windows custom controls (BWCC).
- **VBX**
Choose if your application uses Visual Basic controls.
- **VDBT programmatic access***
Choose if your application uses the programmatic functionality provided with the Visual Database Tools.
- **VDBT with VBX***
Choose if your application uses the Visual Database Tools with Visual Basic controls.
- **VDBT with OLE automation***
Choose if your application uses the Visual Database Tools from OLE automation.
- **ODBC**
Choose if your application requires Microsoft Open Database Connectivity drivers. InstallShield Express supports ODBC 2.1 for 16-bit and ODBC 2.5 for 32-bit. When you select this object, the **ODBC Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. ODBC settings are discussed in detail in the Selecting Individual ODBC Drivers section later in this chapter.

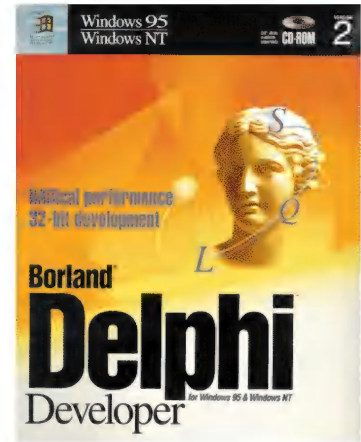
The BDE/IDAPI and SQL Links entries have their own customizable settings, which are explained later in this section. After making your selections, click **Finish** to accept your settings, or **Cancel** to reject them.

If you want to see which files are added or view details of any of the files selected in the **Specify InstallShield Objects for Borland C++** dialog box, click the **Advanced** tab.

Delphi 2.0 Objects

InstallShield Objects for Delphi include support for the Borland Database Engine, SQL Links, ReportSmith Runtime Viewer, and ODBC.

Note If your application does not require any of these add-ons, do not select any of the InstallShield Objects. They will add unnecessary files to your setup.



When you select a Delphi object, Express creates the System Files - WinSysDir group and adds the necessary files and registry/system-file changes for the object selected.

Selecting Delphi Objects

To select the objects to be included with your setup, click the **General Options** push button in the Setup Checklist to open the following dialog:

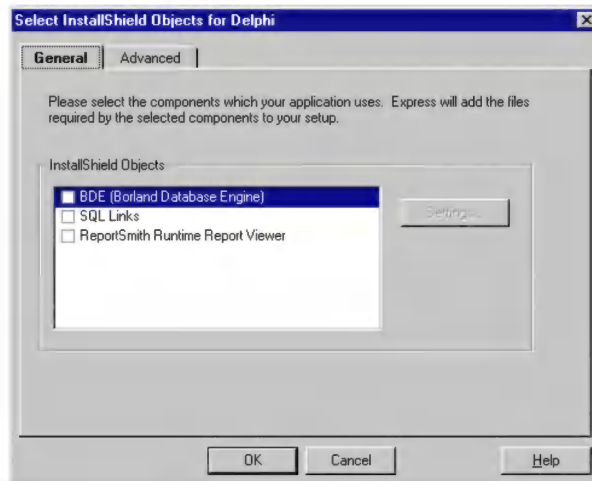


Figure 4.3. Delphi Objects dialog box.

In the **Select InstallShield Objects for Delphi** dialog box, select as many of the following as apply:

- **BDE/IDAPI**
Choose if your application uses the Borland Database Engine. Selecting this check box launches the BDE Settings dialog boxes. This process is covered in detail in the BDE/IDAPI Settings section later in this chapter.
- **SQL Links**
Choose if your application requires SQL links to various database servers. Selecting this check box launches the SQL Links Settings dialog, which is covered in detail in the SQL Links Settings section later in this chapter.
- **ReportSmith Runtime Report Viewer**
Choose if your application uses the ReportSmith run-time module. This process is covered in step-by-step detail in the ReportSmith Settings section below.
- **ODBC**
Choose if your application requires Microsoft Open Database Connectivity drivers. InstallShield Express supports ODBC 2.1 for 16-bit and ODBC 2.5 for 32-bit. When you select this object, the **ODBC Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. ODBC settings are discussed in detail in the Selecting Individual ODBC Drivers section later in this chapter.

ReportSmith Settings

The **ReportSmith Settings** dialog box allows you to install the necessary database connections to run ReportSmith, as well as specify any necessary parameters for your connections. Simply supply the requested information in each of the three dialog boxes and let Express do the work for you.

Note This object is only functional if the BDE object has been selected.

Step 1

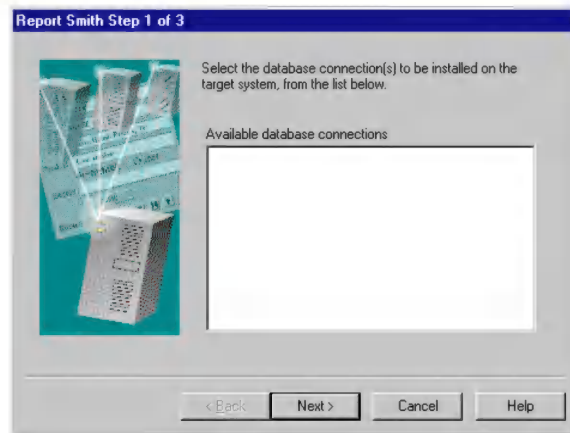


Figure 4.4. Step 1 of ReportSmith dialog box.

The first **ReportSmith** dialog box displays a checklist of the connections that exist on your system. Check the boxes next to the connections that you want to include with your application. When you have selected your connections, click **Next** to continue on to the next step, or **Cancel** to reject your settings.

Step 2

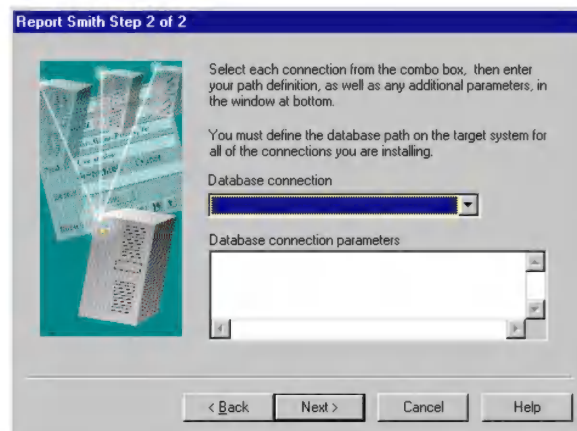


Figure 4.5. Step 2 of ReportSmith dialog box.

To define a parameter for your ReportSmith connection:

- In the **Database connection** field, select the connection for which you want to define a parameter.
- In the **Database connection parameters** window, define the path on the target system for each connection. Use Express Directory Specifiers to indicate the path on the target system, and follow the keyword=path format.

For example, if the keyword for your database is DataFilePath and the file is to be copied to the DATA subdirectory of your main installation directory, you would enter the following:

```
DataFilePath=<INSTALLDIR>\DATA
```

Place each parameter on a separate line in the window. Do **not** use any additional delimiter characters (such as a semi-colon).

Note You must enter the target path for your database file in the proper format or your connection will not be installed on the target system.

When you have defined all of the parameter for each alias, click **Next** to go to step 3, **Back** to return to step 1, or **Cancel** to reject your settings.

Step 3

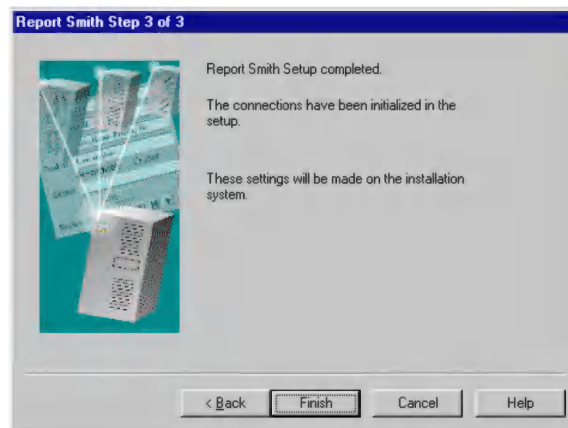


Figure 4.6. Step 3 of ReportSmith dialog box.

The final dialog box informs you that the settings you have specified have been initialized, and Express will configure the target system if you accept the settings. Click **Back** if you want to go back and review or change your settings, or click **Finish** to accept them.

If you want to see which files are added or view details of any of the files selected in the **Specify InstallShield Objects for Delphi** dialog box, click the **Advanced** tab.

Paradox 7.0 Objects

Paradox 7

InstallShield Objects for Paradox 7 include support for the Borland Database Engine (BDE/IDAPI), SQL Links, Paradox Runtime, and ODBC.

If your application does not require any of these add-ons, do not select any of the InstallShield Objects. They will add unnecessary files to your setup.

Note for Version 1.0

The initial release on InstallShield Express Professional 1.0 does not include support for 16-bit InstallShield Objects for Paradox. If you require this support, please check the Express website (URL: <http://www.installshield.com/express>) for an update release which includes 16-bit objects for Paradox.

Selecting Paradox Objects

To begin selecting Paradox objects, click the **General Options** push button in the Setup Checklist. This opens the following window.

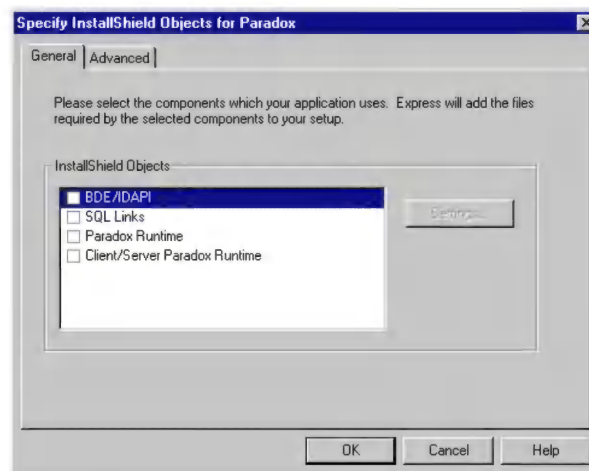


Figure 4.7. Paradox Objects dialog box.

In the **Specify InstallShield Objects for Paradox** dialog box, select the controls that your application requires by clicking the check boxes for as many of the following as apply:

- **BDE/IDAPI**
Choose if your application uses the Borland Database Engine. Selecting this check box launches the BDE Settings dialog boxes. This process is covered in step-by-step detail in the BDE/IDAPI Settings section later in this chapter.
- **SQL Links**
Choose if your application requires SQL links to various database servers. Selecting this check box launches the SQL Links Settings dialog. This process is covered in detail in the SQL Links Settings section later in this chapter.
- **Paradox Runtime**
Choose if your application utilizes the stand-alone version of Paradox Runtime.
- **Client/Server Paradox Runtime**
Choose if your application utilizes the network version of Paradox Runtime.
- **ODBC**
Choose if your application requires Microsoft Open Database Connectivity drivers. InstallShield Express supports ODBC 2.1 for 16-bit and ODBC 2.5 for 32-bit. When you select this object, the **ODBC Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. ODBC settings are discussed in detail in the Selecting Individual ODBC Drivers section later in this chapter.

The **Paradox Runtime** and **Client/Server Paradox Runtime** objects can be added by simply clicking their check boxes. You **must** select either the **Paradox Runtime** or **Client/Server Paradox Runtime** object for Express to include the necessary runtime files for your application.

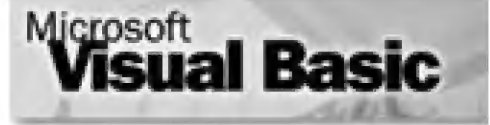
Note SQL Links and Client/Server Paradox Runtime may only be redistributed by registered users of Paradox 7 Client/Server Edition. See Paradox Runtime Developer Help for licensing information or call 800-551-8188 to order Paradox 7 Client/Server. Outside North America, call +1 408-431-1000 to locate your local Borland reseller.

When you select a Paradox object, Express creates the System Files - WinSysDir group and adds the necessary files and registry/system-file changes for the object selected.

Click **OK** to add the **Paradox Runtime** and/or **Client/Server Paradox Runtime** objects or **Cancel** to exit the dialog box without adding any InstallShield Objects.

If you want to see which files are added or view details of any of the files selected in the **Specify InstallShield Objects for Paradox** dialog box, click the **Advanced** tab.

Visual Basic 4.0 Objects



InstallShield Objects for Microsoft Visual Basic 4.0 include support for DAO/Jet, ODBC, and many other popular controls.

InstallShield Express, you can easily add these components to your installation by either of the following methods:

- Allowing Express to automatically review your Visual Basic project file and determine which objects are necessary.
- Selecting the InstallShield Objects to be included by checking the individual components.

Note Express automatically registers all OCX controls containing the “OLE SelfRegister” string in the version resource. This automatic handling of self-registering files applies for **all** development environments. Refer to Self-registration of .DLLs and .OCXs in Chapter 7, “Frequently Asked Questions” for more details.

You can also run an automatic review and modify Express’s selections by hand in the **Select InstallShield Objects for Visual Basic** dialog box.

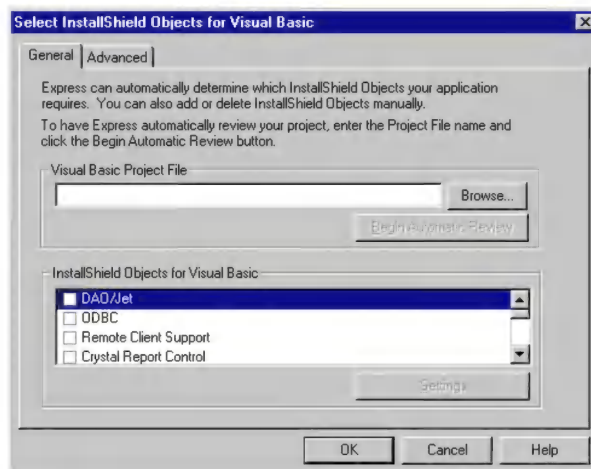


Figure 4.8. Visual Basic Objects dialog box.

Note Regardless of which method you choose and what options you select, if you click **OK** when closing this dialog box, Express automatically creates the VB RUNTIME 32 BIT (or 16 BIT) group and adds Visual Basic run-time files to it. If you do not want to include these files with your setup, click **Cancel** to exit the dialog box. Once the files have been added, you must go to the **Groups and Files** dialog box and delete the group.

Automatically reviewing Visual Basic Project File

Express can review your project file and select the InstallShield Objects required for your application. To have Express review your project, enter your application's project file name in the **Visual Basic Project File** field and click the **Begin Automatic Review** push button. You can search for the project file by clicking the browse (...) push button. This launches the **Open** dialog box, which allows you to select your file.

Express searches your project file to determine its dependencies, then selects the appropriate InstallShield Objects for your application. If your project requires any objects, Express creates the System Files - WinSysDir group and adds the appropriate files to the group.

Note Express does not automatically determine dependencies for .VBX files.

Both the VB RUNTIME 32/16 BIT and System Files - WinSysDir groups have <WINSYSDIR> as their default destination directory.

If you want to view details of any of the files selected in the **Select InstallShield Objects for Visual Basic** dialog box, click the **Advanced** tab.

Selecting Individual InstallShield Objects for Visual Basic

If you prefer, you can select each InstallShield object manually. In the EXE Type box, click the appropriate radio button to specify whether your executable is 32-bit or 16-bit. Then, in the list to the right, select the options you want to include with your application.

Select the check boxes for as many of the following options as apply:

- **DAO/J et**
Choose if your application requires Data Access Object/J et files. When you select this object, the **DAO Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. By default, all drivers are selected for inclusion in the Available Drivers window.
- **ODBC**
Choose if your application requires Microsoft Open Database Connectivity drivers. InstallShield Express supports ODBC 2.1 for 16-bit and ODBC 2.5 for 32-bit. When you select this object, the **ODBC Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. ODBC settings are discussed in detail in the Selecting Individual ODBC Drivers section later in this chapter.
- **Remote Client Support**
Choose if your application uses remote client support controls.
- **Crystal Report Control**
Choose if your application uses the Crystal Reports database report writer.
- **Common Dialog Control**
Choose if your application calls the Visual Basic common dialog controls.
- **Data Bound Grid Control**
Choose if your application calls the Visual Basic data bound grid controls.
- **Data Bound List Control**
Choose if your application calls the Visual Basic data bound list controls.
- **Grid Control**
Choose if your application calls the Visual Basic grid controls.
- **Animated Button Control**
Choose if your application uses animated button controls.
- **Gauge Control**
Select if your application uses gauge controls.
- **Key State Control**
Choose if your application uses key state controls.
- **Multimedia Control**
Choose if your application uses Microsoft multimedia controls.

- **Comm Control**
Select if your application uses the Visual Basic Communications controls.
- **MAPI Control**
Select if your applications uses the Microsoft Messaging Application Programming Interface (MAPI) controls.
- **Masked Edit Control**
Choose if your application uses Microsoft Masked Edit controls.
- **Outline Control**
Choose if your application uses outline controls.
- **Picture Clip Control**
Choose if your application uses PicClip controls.
- **Spin Button Control**
Choose if your application uses the Spin Button controls.
- **3D Controls**
Choose if your application uses Visual Basic user-interactive 3D controls such as 3D check box, 3D command button, 3D frame, etc.
- **Graph Control**
Choose if your application uses the Graph control.
- **Windows Common Controls**
Choose if your application requires access to the Visual Basic Windows Common controls.
- **Rich Textbox Control**
Choose if your application uses the Microsoft Rich Textbox control.
- **Tabbed Dialog Control**
Select if your application uses the Microsoft Tabbed Dialog control.
- **Remote Data Control**
Select the Remote Data control if your application requires high-speed access to ODBC data sources such as Microsoft SQL Server and ORACLE.

If you select any of the above controls, Express creates the System Files - WinSysDir group and adds the appropriate files to the group.

If you want to see which files are added or view details of any of the files selected in the **Specify InstallShield Objects for Visual Basic** dialog box, click the **Advanced** tab.

Visual C++ 4.1 Objects



InstallShield for Microsoft Visual C++ 4.1 include support for DAO/J et, MFC 4.0, ODBC, and other popular controls.

Note If your application does not require any of these add-ons, do not select any of the InstallShield Objects. They will add unnecessary files to your setup.

Selecting Visual C++ Objects

In the **General** tab of the **Select InstallShield Objects for Visual C++** dialog box, select the check boxes for as many of the following as apply:

- **DAO/J et**
Choose if your application requires data access object support. Selecting this object and clicking the **Settings** push button launches the DAO dialog box, which allows you to individually select specific drivers. By default, all drivers appear as selected in the Available Drivers window.
- **MFC**
Choose if your application calls the Microsoft Foundation Class (MFC) Library.
- **ODBC**
Choose if your application requires Microsoft Open Database Connectivity drivers. InstallShield Express supports ODBC 2.1 for 16-bit and ODBC 2.5 for 32-bit. When you select this object, the **ODBC Settings** dialog box will open. This dialog allows you to individually select which drivers you want to include with your installation. ODBC settings are discussed in detail in the Selecting Individual ODBC Drivers section later in this chapter.
- **Grid Control**
Choose if your application uses grid edit controls.
- **Key State Control**
Choose if your application uses key state controls.
- **Microsoft Comm Control**
Choose if your application uses Microsoft Communications controls.

- **Microsoft Masked Edit Control**
Choose if your application uses Microsoft Masked Edit controls.
- **Microsoft Multimedia Control**
Choose if your application uses Microsoft Multimedia controls.

If you select any of the above controls, Express creates the System Files - WinSysDir group and adds the appropriate files to the group.

If you want to see which files are added or view details of any of the objects selected, click the **Advanced** tab of the **Select InstallShield Objects for Visual C++** dialog box.

Advanced Settings

The **Advanced** tab of each InstallShield Objects dialog allows you to see which files Express added to your setup and view the details (date, size, and version, if applicable) of the files. You can reveal or hide the files in a group by clicking on the +/- icon next to the group name. To view the details of a particular file, highlight the appropriate filename in the list box.

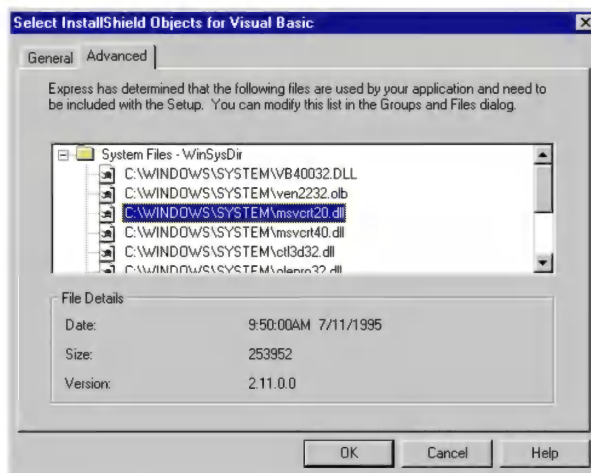


Figure 4.9. Displaying file details in the Advanced tab.

The **Advanced** tab of the **InstallShield Objects for Visual C++** dialog box is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information.

If you need to add additional InstallShield Objects for Visual C++, click the **General** tab. If you need to delete files from the group, you can do so in the **Groups and Files** dialog box.

BDE/IDAPI Settings

The BDE/IDAPI Settings dialog boxes allow you to create BDE aliases, as well as specify any additional parameters for your alias. They also allow you to save the alias settings (.CFG file) so that they can be used by both 16- and 32-bit applications.

When you select the **BDE/IDAPI** check box in the InstallShield Objects window, Express launches the **BDE Installation Type** dialog.

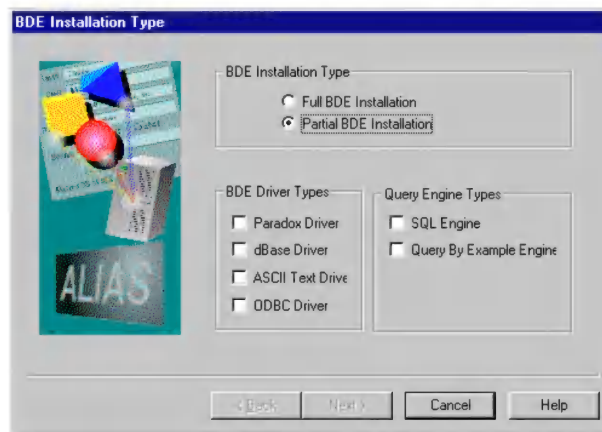


Figure 4.10. BDE Installation Type dialog box.

Use the **BDE Installation Type** dialog box to specify whether you want to install all or part of BDE with your setup.

- If you select the **Full BDE Installation** radio button, all of the available drivers and query engines on your system are added to your setup. Therefore, the check boxes in the BDE Driver Types and Query Engine Types windows remain inactive. You can continue to the next step by clicking the **Next** push button.
- If you only want to install part of BDE, select the **Partial BDE Installation** radio button. Then, select the check boxes corresponding to the BDE driver(s) and query engine(s) you want to include. When you finish, click **Next** to continue on to the alias settings or **Cancel** to reject your settings.

Regardless of whether you selected a full or partial BDE installation, the **BDE Alias** dialog box is automatically displayed after you complete the **BDE Installation Type** dialog. You must define all aliases which your application requires in the **BDE Alias** dialog.

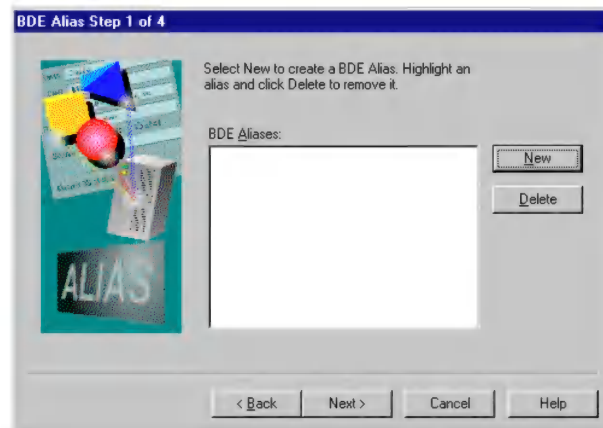


Figure 4.11. Step 1 of BDE Alias dialog box.

Step 1

Click the **New** push button to launch the **BDE Alias Name** dialog box.

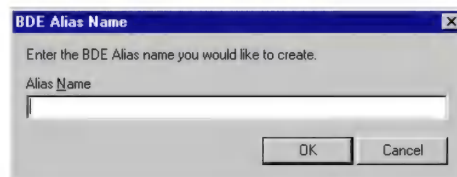


Figure 4.12. BDE Alias Name dialog box.

In this dialog box, type the desired alias in the **Alias Name** field. Click **OK** to enter the alias, or **Cancel** to reject it.

The specified aliases will be listed in the BDE Aliases window. To delete an alias, highlight its name in the window and click the **Delete** push button. When you finish adding all of the aliases you want to include, click **Next** to continue on to Step 2, **Back** to return to the **BDE Installation Type** dialog box, or **Cancel** to reject your settings.

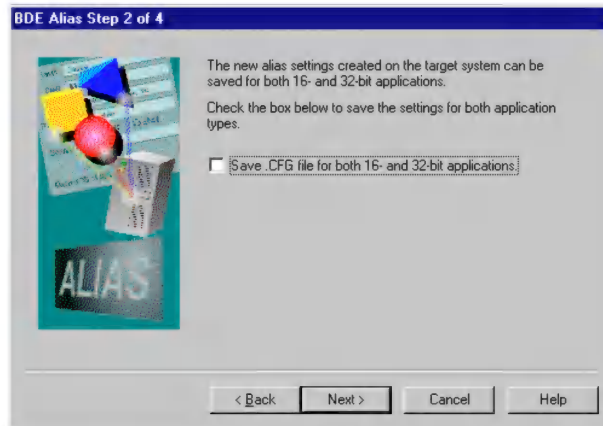
Step 2

Figure 4.13. Step 2 of BDE Alias dialog box.

This step allows you to save the .CFG file containing your alias settings so that both 16-bit and 32-bit applications can use them. If you select the check box, Express automatically makes the necessary modifications on the target system. Click **Next** to go to step 3, **Back** to return to the first step, or **Cancel** to reject your settings.

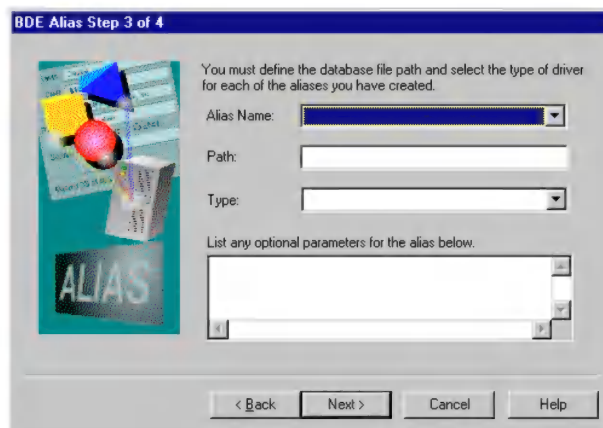
Step 3

Figure 4.14. Step 3 of BDE Alias dialog box.

In Step 3, you must define the parameters for all of your BDE aliases. Use the BDE Configuration utility to determine which parameters each alias requires

To define a parameter for your BDE/ODAPI alias:

- In the **Alias Name** drop-down list, select the alias for which you want to define a parameter.
- Enter target path, not including the filename, of your the database file in the **Path** field. Use Express directory specifiers in this field to indicate the destination directory location of your database file. (Your entry in this field will generally be the same as the value in the Destination Directory field of the Groups and Files dialog for the group containing the database file.)
- In the **Type** field, select the database driver.

Note You must specify an alias name, a target path, and a driver type for your database file or your alias will **not** be created on the target system.

- Enter any additional parameters in the window at the bottom of the dialog box. Parameters must have the following format:

Keyname=Value

Place each parameter on a separate line. Do **not** use any additional delimiter characters (such as a semi-colon). When defining a path in this window, use Express directory specifiers to indicate the location on the target system.

When you have defined all of the parameters for each alias, click **Next** to go to step 4, **Back** to return to step 2, or **Cancel** to reject your settings.

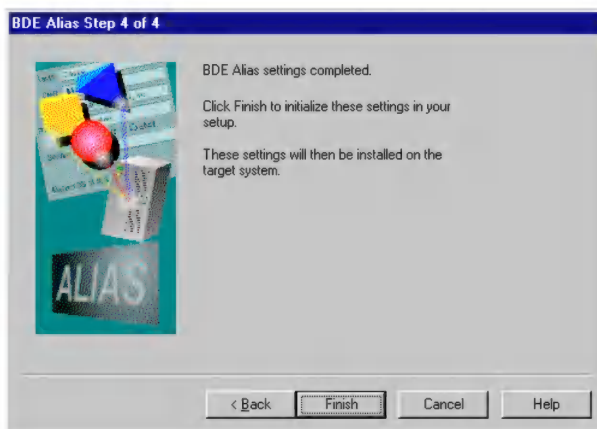
Step 4

Figure 4.15. Step 4 of BDE Alias dialog box.

The final dialog box informs you that the settings you have specified have been initialized, and Express will configure the target system if you accept the settings. Click **Back** if you want to go back and review or change your settings, or click **Finish** to accept them.

SQL Links Settings

When you select the **SQL Links** object, Express automatically displays the SQL Links dialog box with all available drivers selected.

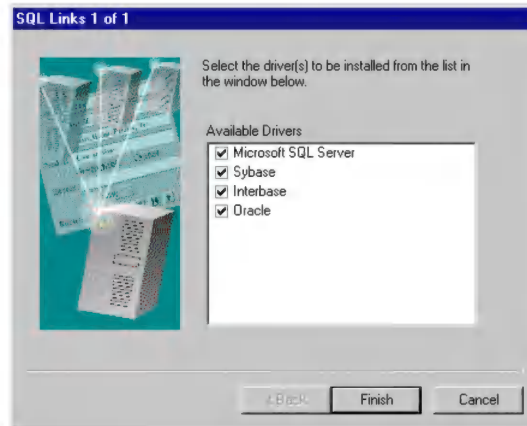


Figure 4.16. SQL Links dialog box.

Uncheck the boxes next to the drivers you do not want installed. Click **Back** if you want to go back and review or change your settings, or click **Finish** to accept your selection of drivers and return to the **General** tab.

Note If you select the SQL Links object, you must select the BDE (Borland Database Engine) object as well.

Selecting Individual ODBC Drivers

When you select the ODBC object, Express automatically displays the ODBC Settings dialog, which allows you to select and configure the drivers and data sources you want to install on your customer's system.

Step 1

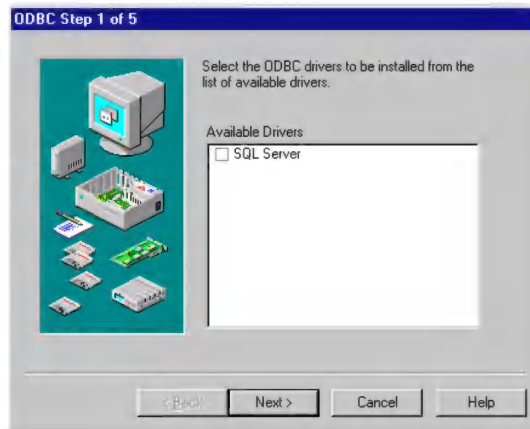


Figure 4.17. Step 1 of ODBC dialog box.

The list displayed in the Available Drivers window is dynamically created based on the drivers found on your system. Select the check box for each ODBC driver to be added to your installation. When have selected the drivers, click **Next** to continue on to Step 2 or **Cancel** to reject your settings and end the process.

Step 2

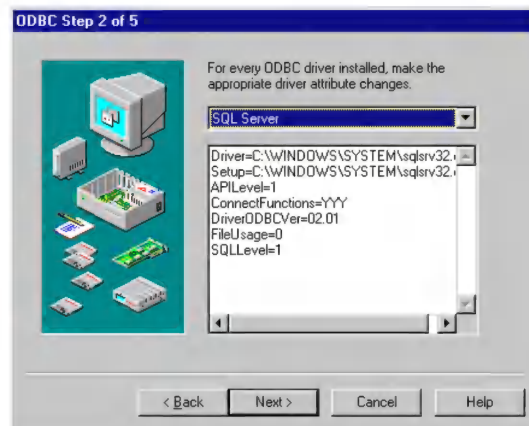


Figure 4.18. Step 2 of ODBC dialog box.

Clicking the down arrow at the end of the upper field in this dialog box reveals a list of the ODBC drivers selected in the previous dialog box. The attributes for the displayed driver appear in the window directly below. You can modify the driver information for each driver as needed by selecting the driver from the list and changing the necessary information. When you finish modifying driver attributes, click **Next** to continue on to Step 3, **Back** to return to Step 1, or **Cancel** to reject your settings and end the process.

Step 3

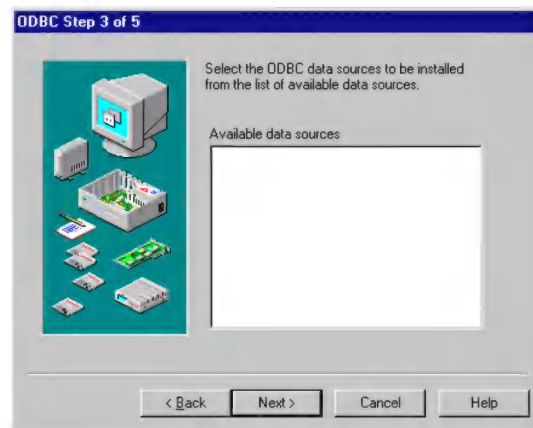


Figure 4.19. Step 3 of ODBC dialog box.

The Available data sources window displays all data source names currently defined on your machine as reported by the ODBC Administrator. Select each data source to be installed on your customer's system by clicking the corresponding check box. After selecting the data sources to install, click **Next** to continue on to Step 4, **Back** to return to Step 2, or **Cancel** to reject your settings.

Step 4

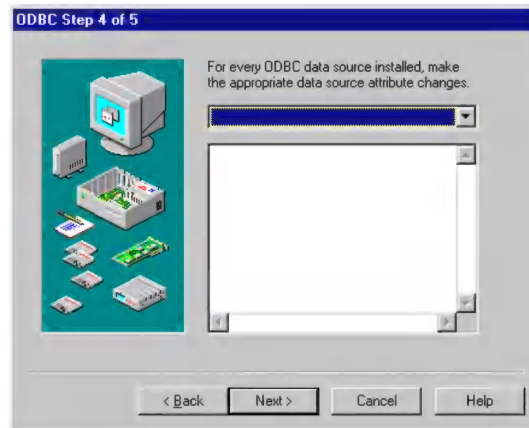


Figure 4.20. Step 4 of ODBC dialog box.

Using the same method described in Step 2, you can modify the attributes for the displayed data source. After making any data source attribute modifications, click **Next** to continue on to Step 5, **Back** to return to Step 3, or **Cancel** to reject your settings and end the process.

Step 5

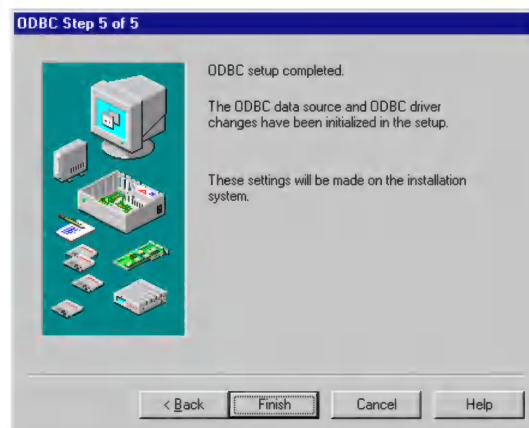


Figure 4.21. Step 5 of ODBC dialog box.

The final dialog box informs you that the settings you have specified have been initialized, and Express will configure the target system if you accept the settings. Click **Back** if you want to go back and review or change your settings, or click **Finish** to accept them.

Chapter 5

InstallShield Extensions

InstallShield Express enables you to create an installation that executes external functions before or after the file transfer process. Your installation can call a .DLL function or run an .EXE file from a point within the installation you select. For example, you may want your installation to launch your application automatically when the setup is complete. InstallShield Extensions provide this functionality.

Extensions are excellent for those projects with very special needs, but they can be tricky to implement. The ability to access an external program during the installation process gives you increased power, but if you do not call the extension correctly, the entire installation can fail. If you are attempting to create an InstallShield Extensions, please carefully read **all** of the documentation provided (both printed and online) before adding the extension to your installation.

Contents

Introduction.....	5-2
Adding an Extension.....	5-2
Calling a Custom .DLL	5-4
Running an .EXE File.....	5-5
Saving Space on Disk 1 for Uncompressed Files	5-7

Introduction

Before you add extensibility to your installation, keep in mind a few basic instructions. Failure to follow these can cause your installation to fail.

1. Make sure that the file being accessed through the extension is accessible on the customer's system when you call the extension.
2. Whenever possible, launch the file from your main installation directory on the target system.
3. Do not attempt to launch a file before it is installed.

If you invoke an extension before the requested file is installed on your customer's system, the extension will not run unless the file was included in _SETUP.LIB or on the first disk of your distribution set.

Adding an Extension

Regardless of the type of extension to be called, the initial steps to adding an extension to your installation are as follows:

1. To open the InstallShield Extensions dialog, select the **InstallShield Extensions** entry from the Setup Checklist.
2. The **Express Extensions** dialog box appears.

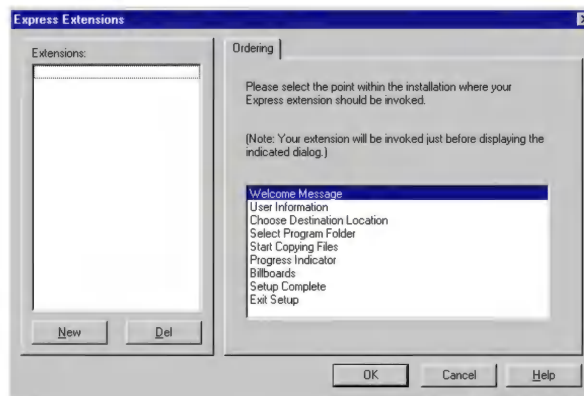


Figure 5.1. The initial Express Extensions dialog box.

The Extensions window on the left side of the dialog displays the files currently selected to be called during the installation.

3. Click the **New** push button beneath the Extensions window to open the **New Extension** dialog box.

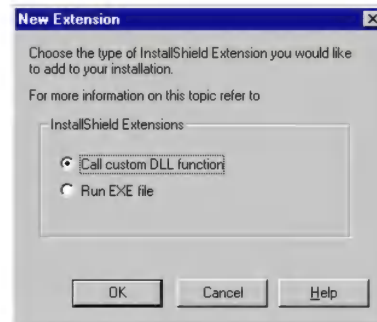


Figure 5.2. The New Extensions dialog box.

Select the appropriate radio button depending on whether you want to call a custom .DLL or launch an executable file. Click the **OK** push button enter your selection and close the dialog box

4. Highlight one of your Express user dialogs to select the point within your installation when you want your extension to be launched. The extension will run **BEFORE** the highlighted dialog.

Note If you have already included an extension and are creating an additional one, do not highlight the location of the new extension until you have added it. If you select a location before clicking the **New** push button, you will change the order of the existing extension which is currently highlighted

5. Once you have selected the point within the setup at which your extension will run, click the **Settings** tab and enter the settings for the appropriate type of extension.

The next step in the process depends on the type of extension you just added. If you selected the **Call custom DLL function** radio button, refer to the Calling a Custom .DLL section of this chapter. If you selected the **Run EXE file** radio button, refer to the Running an .EXE File section of this chapter.

Calling a Custom .DLL

To calling a custom .DLL from within your installation, you must first add the .DLL extension. For more information on this process, refer to the Adding an Extension section earlier in this chapter.

The next step is to enter the settings for the .DLL into the fields on the **Settings** tab of the **Express Extensions** dialog box.

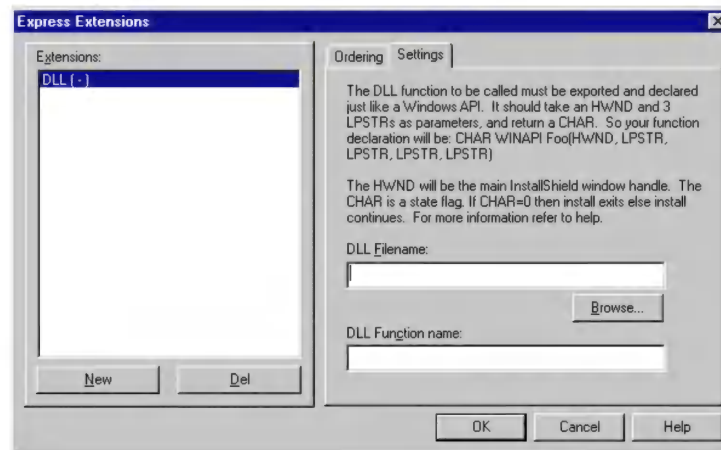


Figure 5.3. Custom .DLL settings.

1. Enter the name of the .DLL in the **DLL Filename** field. Clicking the **Browse...** button opens the Setup Files Browser. You can enter the .DLL filename by highlighting it in the File Groups window and clicking the **OK** push button. You can also use the Express directory specifiers <SUPPORTDIR> or <SRCDIR> with the selected .DLL in conjunction with certain Disk Builder options. For example, an entry of <SRCDIR> MYDLL.DLL could be used if MYDLL.DLL is located uncompressed on disk 1. For more information on Disk Builder options, refer to Chapter 8, "Express Dialog Boxes."

The Extensions window updates the highlighted extension automatically to include the .DLL name.

2. Enter the name of the function you wish to call in the **DLL Function name** field. If you are experienced in using Windows APIs, this function call will be familiar to you. Your .DLL extension must be declared and exported in the same way as a Windows API. Enter your .DLL's function

declaration in the format

```
CHAR WINAPI Foo(HWND, LPSTR, LPSTR, LPSTR, LPSTR)
```

The following parameters will be passed to your .DLL:

- Parameter 1 is the main InstallShield window handle.
 - Parameter 2 is the source directory <SRCDIR>.
 - Parameter 3 is the support directory <SUPPORTDIR>.
 - Parameter 4 is the main target directory <INSTALLDIR>.
 - Parameter 5 is a null field, reserved for future use.
 - CHAR is a state flag signaling the completion of the routine. If your function return CHAR=0 from your .DLL, then the installation exits. If it returns any other value, the installation will continue.
3. After entering the .DLL extension information, click the **OK** push button to register these settings.

Express enables your application to store uncompressed files on the first disk of the distribution set. If your extension needs to access an uncompressed file before the file transfer process begins, refer to the Saving Space on Disk 1 for Uncompressed Files section later in this chapter for detailed instructions.

Running an .EXE File

To run an .EXE file from within your installation, you must first add the .EXE extension. For more information on this process, refer to the Adding an Extension section earlier in this chapter.

The next step is to enter the settings for the .EXE into the fields on the **Settings** tab of the **Express Extensions** dialog box.

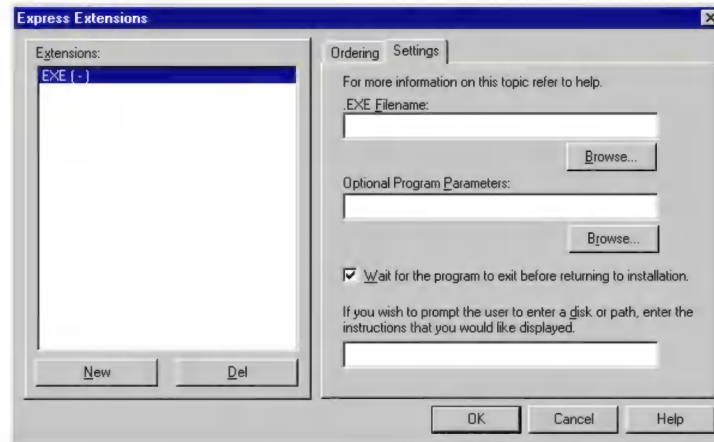


Figure 5.4. Configuring .EXE extension settings.

1. Enter the name of the .EXE in the **.EXE Filename** field. Clicking the **Browse...** button opens the Setup Files Browser. You can enter the .EXE filename by highlighting it in the File Groups window and clicking the **OK** push button.

Note The executable file to be launched must have a window handle. Express requires this handle to return from the extension to the installation. A DOS executable will not function correctly as an InstallShield Extension.

The Extensions window updates the highlighted extension automatically to include the .EXE name.

2. Enter any optional file parameters needed for this extension in the **Optional Program Parameters** field. You can manually enter the parameters or click the **Browse...** button to open the Setup Files Browser.
3. Select the **Wait for the program to exit before returning to installation** check box if you want your setup to wait for the .EXE file to finish executing before continuing the installation.
4. If it will be necessary to prompt the customer to insert a disk to run the .EXE file, Express enables you specify the disk to be inserted by name. Enter the disk label text to appear in the prompt by typing it into the bottom field on the **Settings** tab.

Express enables your application to store uncompressed files on the first disk of the distribution set. If your extension needs to access an uncompressed file before the file transfer process begins, refer to the next section, Saving Space on Disk 1 for Uncompressed Files for detailed instructions.

Saving Space on Disk 1 for Uncompressed Files

When you are creating the disk images for your setup project, one of the options available to you is reserving space on the first disk for uncompressed files. One of the purposes of this feature is to allow uncompressed file accessibility for InstallShield Express Extensions. While we encourage you to have your extension call files once they are decompressed on the target system, there are occasions when you might want to call an extension before the file transfer process begins. In these situations, you will need the file for the extension to be in an uncompressed form on the distribution disk.

To add uncompressed files to the first disk image of your installation, perform the following steps.

1. Click the **Disk Builder** push button on the Setup Checklist. The **Run Disk Builder** dialog box appears.
2. Click the **Settings** tab to display the Extra files for disk 1 window.

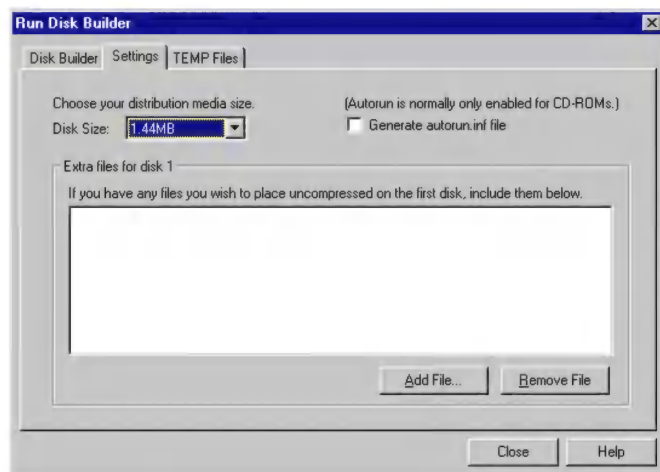


Figure 5.5. The Extra files for disk 1 window.

3. Click the **Add File...** push button to open the file browser. Select the .DLL or .EXE file used in the extension and click the **Open** push button to add it to the Extra files for disk 1 window. The Extra files for disk 1 window updates to display the full path of the selected file.
4. If you are ready to build the disk images, click the **Disk Builder** tab and click the **Start Build** push button. For more information on creating your disk images, refer to Chapter 2, "A Quick Tutorial."
5. Click the **Close** push button to close the Run Disk Builder dialog box.

If you want to remove a file from the Extra files for disk 1 window, simply highlight the file and click the **Remove File** push button.

Note If you would like additional information regarding Express Extensions, please refer to Express' online help and the technical information located on the InstallShield Express website at www.installshield.com/express.

Chapter 6

The Registry

The arrival of Windows 95 also brought the introduction of the Windows registry to the common user's desktop. This chapter gives an overview of the registry's function and its operation, and provides an explanation of how Express enables you to use the registry in your installations.

Contents

What is the Registry?.....	6-2
Keys and Values.....	6-5
Your Installation and the Registry	6-8
Creating Custom Keys and Values	6-10
Removing Custom Registry Keys.....	6-14
Registry Editor	6-15

What is the Registry?

The registry is the Windows 95 and Windows NT database for storing all system and application information. It is a single location for all information needed to set up and configure the system. The registry stores the following types of information:

- Application information (company name, product name, version number, etc.).
- Program names associated with file extensions.
- Command line to be executed when the customer opens the file from a shell application.
- Path information for the application.
- Uninstall information.

Understanding how the registry operates is vital to creating an optimal installation. InstallShield Express enables your software to write its settings to your customer's registry during the installation process. With a little planning, you can help ensure that your product registers properly on your customer's system.

What Does the Registry Do?

The main benefit of storing all of the information solely in the registry is that the system need not be configured through the use of AUTOEXEC.BAT, CONFIG.SYS, and .INI files. According to the Windows 95 Setup Guidelines, your application should no longer record information in the WIN.INI file. Instead, this information should be stored in the registry.

The registry is similar to the individual .INI files used for Windows 3.1 applications. In fact, the registry replaces the function of most .INI files needed for Windows 3.1 applications. .INI files used bracketed section headings followed by entries.

```

[Groups]
Group1=C:\WIN95\PROGRAMS.GRP
Group2=C:\WIN95\ACCESSOR.GRP
Group3=C:\WIN95\DESKTOP.GRP
Group4=C:\WIN95\SYSTEMTO.GRP
Group5=C:\WIN95\DOCUMENT.GRP
Group6=C:\WIN95\GAMES.GRP
Group7=C:\WIN95\MULTIMED.GRP
Group8=C:\WIN95\MAIN.GRP

[Settings]
Order= 2 3 4 5 6 1 7 8
Window=28 22 628 454 1

```

Figure 6.1. Sample .INI file.

As you can see, you were a bit limited in the organization of an .INI file. For example, there was really no way to create a subheading in an .INI file. The registry enables you to do this by creating subkeys, which we will cover in greater detail later in this chapter.

The registry consists of keys and corresponding values. The keys are similar to the bracketed section headings in an .INI file. The values listed under each heading in an .INI file are like the registry values. The registry can store binary data, which is not possible in an .INI file.

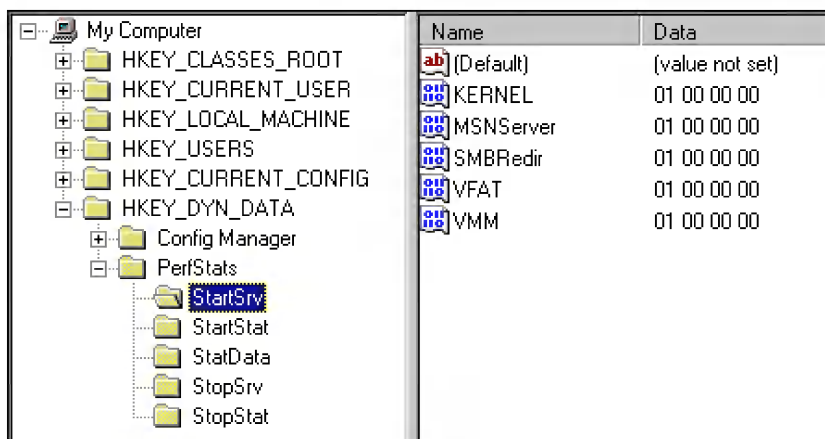


Figure 6.2. A view of the registry.

Notice that the registry is organized as a tree structure, starting with the six standard base keys. The tree structure creates unique access pathways to system and application information. Registry keys denote unique locations for storing information.

While the registry itself is new in Windows 95, the concept did exist in Windows 3.1, which has a registration database file consisting of a single key structure (**HKEY_CLASSES_ROOT**). However, the bulk of system, user, and application related information in Windows 3.1 is stored in various .INI and .SYS files.

Note If you are using Express to create a 16-bit installation to run under Windows 3.1, any registry keys that you create will appear under the **HKEY_CLASSES_ROOT** key, regardless of what you entered in the **Registry - Keys** dialog box.

How does the registry work?

Before Windows 95 and Windows NT 3.51, the system needed to check multiple configuration files in order to perform basic tasks like starting, accessing a network, and running applications. With Windows 95 and Windows NT 3.51 and higher, the operating system checks only one location before executing any task — the registry.

Let's take a look at the different ways the system accesses the registry to understand the possible impact on your installation.

- **Hardware and Application Configurations**

Each time a setup program runs or a Plug and Play device is added or removed under Windows 95, the registry receives the applicable hardware configuration data from the Configuration Manager. The registry maintains a list of the detected system hardware and adjusts it every time a change is made to the system. The benefit of this is two-fold. Instead of having to write entries to a CONFIG.SYS or AUTOEXEC.BAT file whenever a new item of hardware is added to the system, the registry tracks these entries on a continual basis. When a hardware item is removed, the appropriate entries are also removed from the registry, automatically freeing up resources for later use.

- **Device Drivers**

The registry communicates with the various device drivers relating to configuration data and load parameters. Device drivers relay system resource information such as DMA channels and hardware interrupts to

the registry. This information is then available to other applications and drivers on the system. In Windows 3.1 applications, this information would usually be located in the device=lines of a CONFIG.SYS file or a SYSTEM.INI file.

- **Administrative Tools**

Whenever the customer makes changes to the Control Panel settings, File Associations, System Policies, or installed software, the changes are reflected in the Registry. This allows changes to be made without actually accessing the Registry Editor, which will be addressed later in this chapter.

As you can see, your installation must declare your application's settings to the registry. The remaining sections of this chapter provide the necessary information to enable your installation to register your software easily and accurately.

Keys and Values

The registry is a database organized with a hierarchy. All registries start with the root key, which is "My Computer." The next level consists of the six base keys.

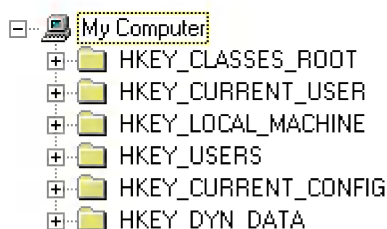


Figure 6.3. The root and base keys.

A key is a named node in the registry tree that can contain a subkey, a value name and value pair, and a default value. When a key has one or more subkeys, a +/- indicator appears next to the folder icon. You can click this indicator to either hide or reveal the subkeys.

Note The various keys in the registry are not individual files and cannot be treated as such. The keys and values comprising the registry are all part of a central database.

The Base Keys

The six base registry keys contain data specific to the system and its users. Notice that each of the base keys begins with “HKEY_”. This indicates that each key is a handle that can be used by a program. The following six keys are common to all registries:

- **HKEY_CLASSES_ROOT**
This key contains user interface information about OLE and association mapping. Without this data, drag-and-drop operations and Windows 95 and Windows NT shortcuts would not be possible. Most of the file extension and file type information contained in WIN.INI resides here. This is the only key that existed in the Windows 3.1 registration database.
- **HKEY_CURRENT_USER**
This key contains data applicable to the current user.
- **HKEY_LOCAL_MACHINE**
This key contains the computer, drivers, and system setting specifications for the system. Information such as the installed hardware, port mapping, and software configurations are stored under this key.
- **HKEY_USERS**
This key contains user-specific and general information about all users who have access to this particular system. Information such as desktop configurations, default application settings, and event schemes are stored under this key. In the case of a multiple user system, the **HKEY_CURRENT_USER** key displays the specific data for the current user.
- **HKEY_CURRENT_CONFIG**
This key is actually a pointer to a branch of **HKEY_LOCAL_MACHINE\Config** that contains the current hardware configuration information for this system, such as the monitor and the printer.
- **HKEY_DYN_DATA**
This key stores the status data for the various dynamic devices attached to the system. This information updates whenever a device is added to or removed from the system. Every device detected by the system has an associated hardware key that retains the specific status values for the device.

You cannot add any additional base keys. Each key you create must therefore be a subkey of an existing key. Each of these base keys can house both values and additional subkeys. The terms “key” and “subkey” are relative to their position. In the registry hierarchy, any key that is below another key can be referred to as a subkey.

Value Entries

Value entries are two-part data structures under a key. Every value consists of a value name and value pair. The value name identifies the value for storage under the key. The value is the data associated with the value name. Figure 6.4. displays the relationship between keys, default values, and named values.

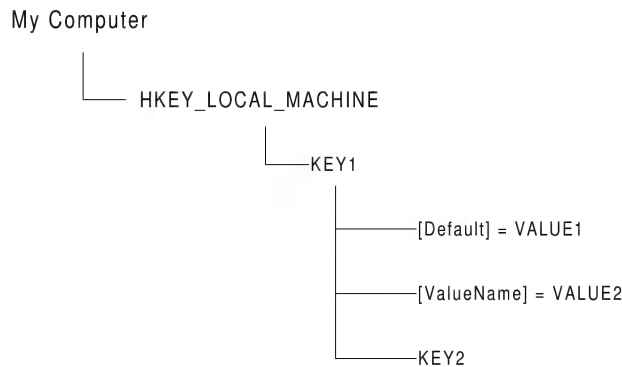


Figure 6.4. The relationship between keys and values.

In Figure 6.4., **KEY1** is a key under the base key **HKEY_LOCAL_MACHINE** containing a default value (**VALUE1**), a value name and setting (**[ValueName] = VALUE2**), and a subkey (**KEY2**).

Registry keys and their values are depicted in the following path name-like format:

[HKEY_LOCAL_MACHINE]\ KEY1
[Default] = VALUE1
[ValueName] = VALUE2

Backslashes (\) separate the keys. The value names and value pairs are entered underneath the entire key expression. The value name appears on the left side of the equal sign and the value on the right. A value cannot exceed 64K.

Your Installation and the Registry

Information about each software application installed or accessible to a Windows 95 or Windows NT 3.51 or higher system is stored in that system's registry. Obviously, if you want your application to install properly on your customer's system, you need to ensure that your application's settings are added to the correct registry keys. InstallShield Express, through the use of default settings and Express directory specifiers, makes it very simple to assign your application's information to the appropriate registry keys. If you complete each of the steps listed in the Setup Checklist, you will have provided the necessary registry information. The following sections take a closer look at the specific registry keys that your setup should address.

Application Information Key

The registry's application information key stores the data critical to the local installation of each application. Express enables your setup to create a unique registry location for your application. Basically, this key is where you notify the registry of your application's name, version number, and your company name. This information is added via the **App Info** tab of the **Set the Visual Design** dialog box in the Setup Checklist. Once added to the registry, your product's application information key will follow this format:

[HKEY_LOCAL_MACHINE]\ SOFTWARE\ <Company>\ <App Name>\ <Version>

Per Application Paths (App Paths) Information Key

Note Per application paths apply to 32-bit applications only. If your application runs in DOS or Windows 3.1 (standard 16-bit mode), the per application paths entry does not apply.

The per application path key, or App Paths key, stores path information which enables the operating system to locate your application's executable files. Prior to the arrival of Windows 95 and Windows NT 3.51, this type of information was stored in the AUTOEXEC.BAT file's SET PATH= statements. According to the Windows 95 Setup Guidelines, new applications should not modify the AUTOEXEC.BAT file. The registry now stores this path information under application-specific keys located under the key **[HKEY_LOCAL_MACHINE]\ SOFTWARE\ Microsoft\ Windows\ Current Version\ App Paths\ <YOURAPP.EXE>**.

The key name **YOURAPP.EXE** is determined by the executable filename you entered in the **Application Information** dialog box. Each executable file installed has a key entry bearing the executable's name in the registry and two value pairs:

- The **[Default]** value of this key contains the fully qualified path of your primary executable file.

Note You must specify a file in the Application Executable field of the **Application Information** dialog box. If you do not, Express will not make the App Paths entry for you.

- The **[Path]** value name enables you to set an additional path by using the **Registry - Values** dialog box to create the **[Paths]** value name under the same key, and add the desired path as the value.

Once this information is registered, the shell provides the application path to ShellExecuteEx when the application is executed. Additionally, this information allows the customer to start your application by selecting **Run** from the Start menu and typing the filename of the executable.

Application Uninstallation Key

The next key that is critical for your application is the application uninstallation key. This is where all information enabling uninstall functionality is stored. Since uninstallation capability is required for all Windows 95 and Windows NT 3.51 and higher applications, it is vital that the registry receives the correct information.

InstallShield Express handles the removal of your application's registry additions through unInstallShield. Unless you specifically clear the **Automatic Uninstaller** check box located on the **Features** tab of the **Set the Visual Design** checklist entry, Express automatically makes the necessary uninstallation registry entries to satisfy Windows 95 Setup Guidelines. As long as you do not add any additional customized registry keys to your setup, unInstallShield can automatically remove all of your application's registry entries. If you have added any customized keys, you also need to set uninstall keys. This topic is covered in greater detail later in this section.

Your application's uninstall data is stored in a key under the **[HKEY_LOCAL_MACHINE]\ SOFTWARE\ Microsoft\ Windows**

Current Version\ Uninstall key. This subkey takes the name of your application and has two value name and value pairs:

- The **[DisplayName]** value contains the information you entered in the **Application Name** field of the **Application Information** dialog box. This name appears in the Install/Uninstall tab of the Windows 95 Add/Remove Programs dialog in Control Panel.
- The value of **[UninstallString]** identifies the exact locations and names of the uninstall executable file and the log file containing uninstall data.

Creating Custom Keys and Values

Express enables you to easily create customized keys for your application to add to your customer's registry. These operations are listed in the Setup Checklist under the **Make Registry Changes** heading. The purpose of this section is to illustrate how you can modify the registry entries to ensure your application installs properly and meets the Windows 95 Setup Guidelines.

Registry - Keys

The **Registry - Keys** tab of the **Make Registry Changes** dialog box allows you to add keys to the registry. This tab is dynamically linked with the **Registry - Values** tab, which lets you add or modify the values of the highlighted key in the **Registry - Keys** tab.

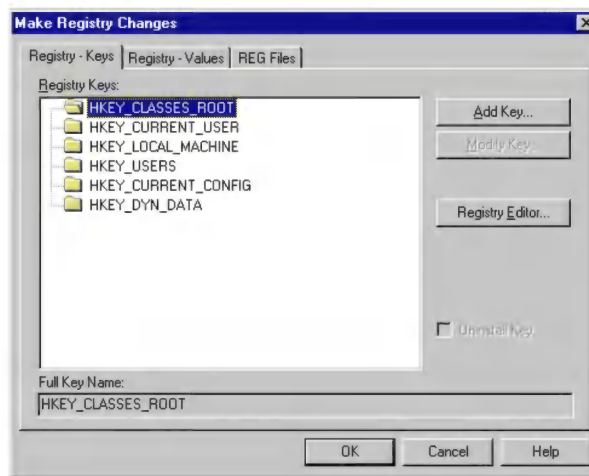


Figure 6.5. The Registry - Keys dialog box.

As we mentioned earlier, you cannot add to or modify any of the six root keys. Each key you create must therefore be a subkey of an existing key. Because you also need to set any new key with a corresponding uninstallation key to comply with Windows 95 guidelines, we recommend that you create the subkey using the format:

\ <Company>\ <Application>

By using your company and application names, you create a unique key. This enables the customer to uninstall the application without deleting information from any other applications.

Note Step-by-step illustrated examples of these procedures are included in Chapter 2, "A Quick Tutorial."

To add a new key:

1. Click the existing key below which you want to create your new key.
2. Click the **Add Key...** push button. The **Registry New Key** dialog box displays.
3. Enter the name you want to give the key in the **New Key** field.

Note You can use Express directory specifiers in your key name. For additional information about directory specifiers, refer to Chapter 3, "Building Your Installation."

4. Click **OK**.

You can enter more than one level of subkeys at once using a backslash to separate them (**Software\ Company\ MyApp\ 3.0\ Help Files**).

To delete a key you have entered:

1. Highlight the key in the **Registry Keys** tab.
2. Press the **Delete** key on your keyboard.

Note When you delete a key that has subkeys, all the subkeys are deleted as well. Refer to the Removing Custom Registry Keys section for information on the potential problems involved in deleting registry keys before deleting any key.

To modify a key you have entered:

1. Highlight the key in the **Registry Keys** tab.
2. Click the **Modify Key...** push button. The **Registry Modify Key** dialog box appears.
3. Retype the name in the **New Key** field.
4. Click **OK**.

The full path and name of the highlighted key display in the **Full Key Name** field, truncated at the right. You can click the **Registry Editor...** push button to launch your registry editor.

Once you have added a key, you can set the value(s) of the key by highlighting it and clicking on the **Value** tab. When you are finished with the registry dialogs, click **OK** to accept your entries or **Cancel** to reject them.

Registry Values

The **Registry - Values** tab of the **Make Registry Changes** dialog box allows you to set value name and value pairs for the key which was highlighted in the **Registry - Key** tab before you clicked the **Registry - Values** tab.

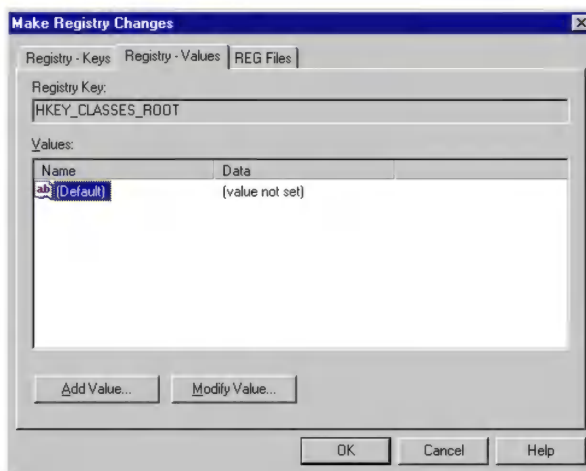


Figure 6.6. The Registry - Values dialog box.

The name of the key being modified displays in the **Registry Key** static text field. If you want to add to or modify values from a different key, simply click the **Registry - Keys** tab, highlight the key you want to update, and click the **Registry - Values** tab again.

To add a value name and value pair:

1. Click the **Add Value...** push button to open the **Registry Value** dialog box.
2. Select one of the radio buttons in the Value Type window to specify whether your data is in string, binary, 32-bit decimal, or 32-bit hexadecimal format.
3. Type the value name in the **Value Name** field and the value (data) in the **Value Data** field. You can use one of the Express directory specifiers in your data.

Note If you are entering a hexadecimal value, do not use spaces in the value. Enter the data as a continuous string of characters.

4. Click **OK** to accept or **Cancel** to reject your entry.

To delete a value name and value pair:

1. Highlight the name of the value name and value pair you want to delete.
2. Press the **Delete** key on your keyboard.

To modify a value name and value pair:

1. Highlight the name of the value name and value pair you want to modify.
2. Click the **Modify Value...** push button. The **Registry Value** dialog box opens with the **Value Type** radio buttons grayed out.
3. Enter the new name or value data in the appropriate field.
4. Click **OK** to accept or **Cancel** to reject your change.

When you finish adding or modifying value name and value pairs for the key, you can click **OK** to accept your changes, **Cancel** to reject them, or the **Keys** tab to select a different key.

Removing Custom Registry Keys

If you do not add any additional registry keys, you do not have to select an uninstall key for your application in order to support unInstallShield. However, in order to comply with the setup guidelines, you must uninstall all registry entries created by the application. Therefore, every key your application adds to the customer's registry must have a corresponding uninstall key for unInstallShield. Select an uninstall key by highlighting the desired key in the **Registry - Keys** dialog box and selecting the **Uninstall Key** check box.

WARNING

Be **very cautious** when selecting your uninstall key. If you select a key which has existing entries on the target system, such as the **HKEY_LOCAL_MACHINE\SOFTWARE** key, and the customer chooses to uninstall your application, **all** of the keys and values under that key, including those from other applications, will be deleted.

Registry Editor

As you have seen from the illustrations in this chapter, Windows 95 has a visual registry viewer called the Registry Editor. The Registry Editor displays the hierarchical structure of the registry database.

You may have noticed that the Registry Editor does not appear on any menu or on the desktop. This is because any changes made to the registry should only be made by experienced users. A mistakenly deleted or modified entry in the registry can corrupt the entire database—a situation no customer ever wants to experience. Whenever possible, you should use the tools available in the Windows 95 Control Panel to make any configuration changes to a system.

To open the Registry Editor, select **Run** from the Start Menu and type `regedit` in the **Open** field, or double-click the Registry Editor icon from the Windows Explorer. If you are in Express, the **Registry - Keys** dialog box also has a **Registry Editor...** launch button.

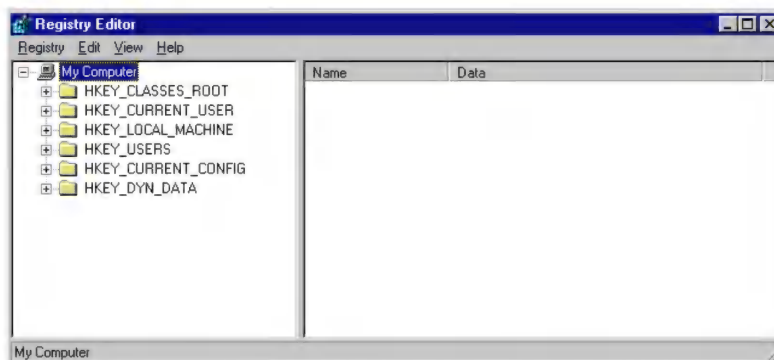


Figure 6.7. The Registry Editor window.

We present this Registry Editor information for educational purposes only. The registry is a huge database not to be altered on a whim. InstallShield Express creates the registry entries for your installation from within Express. You need not access the Registry Editor directly to create your installation. For specific information on modifying the registry through the Registry Editor, please consult the appropriate Microsoft documentation.

Chapter 7

Frequently Asked Questions

This chapter presents some of the more common questions Express users have about InstallShield Express. Please look through these before contacting us for technical support.

We also recommend that you consult the InstallShield Express website at <http://www.installshield.com/express>. A list of frequently asked questions (FAQs) is provided at our website in the Technical Support section. This list is updated daily with new questions we receive and the accompanying answers. If the question you are looking for does not appear in this chapter, check the website's FAQ list.

If you have a question of your own regarding Express, please let us know. You can email a Technical Support Form with your question to us at express@installshield.com. You can find a Technical Support Form by selecting **Technical Support** from Express's **Help** menu. To help us answer your question as quickly as possible, please be as detailed as you with your question. If your question is one commonly asked, we will add it to the FAQ list on our website and include it in the next printing of this user's guide.

The questions and answers in this chapter are grouped by subject matter.

Contents

BDE	7-2
Create Distribution Media.....	7-3
Custom Installations.....	7-3
Disk Builder	7-4
Folders and Icons	7-6
ODBC	7-6
The Registry	7-6
Running the Installation	7-7
Self-registration of .DLLs and .OCXs	7-9
Setup Groups and Components.....	7-10
Uninstallation	7-12

BDE

Why isn't the SQL Server alias " getting installed on the target system? "

The SQL Server alias does not use the PATH parameter. However, all BDE alias definitions require that there be a NAME, PATH, and TYPE parameter in the alias definition. Without a PATH parameter, the alias does not get created during the installation.

In the case of an alias which does not require a path, the path can be set to the <INSTALLDIR> directory specifier. Note that the BDE Configuration utility does allow an alias to be created manually without the PATH. However, the Borland AddAlias installation function (which is used by InstallShield Express) will NOT allow this.

Therefore, when making your alias setting for a SQL Server driver, the PATH should be set to <INSTALLDIR>.

Where will the BDE files be placed during a partial BDE installation? "

All the BDE Engine files (except for the .BLL files and the configuration files) will be placed in the application's main installation directory, <INSTALLDIR>. The .BLL and configuration files will be placed in the Borland common files location: <ProgramFilesDir>\Borland\ Common Files\ BDE.

How do I specify optional parameters for my BDE alias?"

In the Step 3 dialog box of BDE Alias Settings, each optional parameter should be entered in the following format:

KEYNAME=VALUE

Each keyname/value pair should be entered exactly as it is displayed for the alias in the BDE configuration utility, and each pair should be entered on a separate line. No additional formatting or delimiters are necessary or allowed.

For complete directions on adding BDE aliases, refer to Chapter 4, "InstallShield Objects."

Create Distribution Media

How do I create a self-extracting executable file for my setup?

InstallShield Express enables you to create a single-file self-extracting executable for your installation (if you have created the disk images) by simply selecting **Path for a 1 File Installation** in the **Copy to Floppy** dialog box. For complete instructions on this process, as well as an overview of the security options available and uses for this type of installation, refer to Chapter 3, "Building Your Installation."

Custom Installations

How do I change my setup's destination directory?

There are two ways to specify a destination directory in Express:

- **You can specify a destination for each Program Group.**

This can be done in the Express Groups and Files dialog box. Highlight the file group to which you want to assign a destination. Then, in the **Destination Directory** field, choose an Express directory specifier. This specifier can be appended with a "hardcoded" sub-path. An example is <INSTALLDIR>\HelpFiles. Your setup would then create the HelpFiles directory under the default value of the specifier <INSTALLDIR>. It is not recommended that you hardcode this entire entry, as the drive you enter may not exist on your end user's system.

- **You can change the default value of the directory specifier <INSTALLDIR>.**

This means that any Program Group assigned a destination directory of <INSTALLDIR> (along with the uninstall-related files that Express includes with your setup) will be installed to this default directory. Go to the Express **Dialog Boxes** dialog box, highlight **Choose Destination Location**, and click the **Settings** tab. In the field, enter your default installation directory. Once again, do not hardcode this entire entry, but instead use directory specifiers. For example, the entry <WINDISK>\MyAppFiles would cause your setup to create a root directory named MyAppFiles, and this directory would be placed in the drive on which your customer's Windows directory is installed. This

setting will apply even if you don't include the **Choose Destination Location** dialog box in your setup.

Note Do not use the <INSTALLDIR> specifier to indicate the default destination directory in the **Choose Destination Location** dialog box.

For more information on Express Directory Specifiers, please refer to Chapter 3, "Building Your Installation."

Disk Builder

Why is Disk Builder unable to create a directory for my project's disk images ?

Note If you have been seeing an error message from Disk Builder ("Error: Unable to create directory..."), this answer will correct the problem.

When building your project's disk images, Disk Builder uses the name you gave your Express project to create a directory, and saves the images in a path under this directory (for example: MyProjectName\ 144MB\ DISK1\). This new directory is created under the same directory in which your project (.IWZ) file is located. So, if this directory contains a file with the same name as your project file less the .IWZ extension, Disk Builder will fail when trying to create the disk image directory, since the name will already exist.

For example, if you name your project MyApp, it will be saved as MyApp.IWZ to the directory you select. If, within this directory, you already have a sub directory named MyApp (no extension), Disk Builder will then not be able to create a directory called MyApp under which to place your project's disk image directory structure.

To alleviate this problem, please move your project's .IWZ file to a directory in which its name is unique, and then rerun Disk Builder.

Why is Disk Builder completing the disk builder process?

The problem is most likely occurring during file compression. The following is a list of possible causes for a failing file compression:

Not enough space available in the project's directory.

Check to see that you have ample space available within the directory in which you saved your Express project, and that it is not write-protected.

A corrupt compression program caused by a media related problem such as a corrupt hard-disk or diskette.

- Run ScanDisk on the system's hard drive, if available, to see if there are any bad sectors.
- Request a new set of Express diskettes.

A conflict with a concurrently running program such as a screen saver or virus program.

Shut down any concurrently running programs and attempt to rerun the build. This includes any virus checking software which may be running in the background.

Problem with a network drive.

Move your Express project file to a local drive and attempt to rerun the build.

A conflict with an installed driver.

- Modify the AUTOEXEC.BAT and CONFIG.SYS files to load as few drivers as possible and attempt to rerun the build.
- Attempt the build on as many systems as possible, and see if it runs on any other systems. If it does, try to determine what drivers are installed on the problem system(s).

A conflict with the machine's hardware or peripherals.

Remove as many peripherals as possible and attempt to rerun the build.

You can also open our compression program ICOMP.EXE in DOS. This file is located in the directory in which you installed Express. In DOS, type `ICOMP` to see this file's Main Options list. See if you can use this program to compress any other files on your system.

Folders and Icons

How do I assign an icon resource to an executable that I'm installing?

The resource (such as an .ICO or .DLL file) containing the icon that you want to use must have already been dragged and dropped from Explorer into one of your file groups within the **Groups and Files** dialog box.

After you have added the executable in the **General** tab of the **Select Folders and Icons** dialog box, click the **Advanced** tab. Highlight the executable to which you want to assign the icon resource. Then click on the browse (...) button next to the Icon field to open the Setup Files Browser window. Select the desired resource file and click OK.

When you return to the **Advanced** tab, the resource file will be listed in the **Icon** field. Click the **Modify Info** push button to apply the new icon.

ODBC

Why aren't all my ODBC drivers displayed in the ODBC Settings window for my 16-bit installation?

The list of available 16-bit ODBC drivers is read by Express from the ODBCINST.INI file. If you have a 16-bit ODBC driver which is not recorded in this file, then it has not been properly installed in your system.

Note Any 32-bit ODBC drivers (which are logged in the registry) will not be displayed in a 16-bit ODBC setup project.

The Registry

How do I make registry entries using binary and hex value data?

In the **Registry - Keys** dialog, highlight the key to which you want to add a value pair, and then click on the **Registry - Values** tab. Click the **Add**

Value push button to open the **Registry Value** dialog. Type the name in the Value Name field.

To enter binary value data, select the **Binary** radio button in the Value Type window. Next, in the **Value Data** field, type your data in HEXADECEMAL format, with no spaces between digits. For example, a binary entry of 6F05E would, after installation, appear in the registry as 06 f0 5e.

To enter hexadecimal value data, select the **DWORD Hex** radio button in the Value Type window. Then, in the **Value Data** field, type your data in HEXADECEMAL format, with no spaces between digits. For example, a DWORD Hex entry of 6F05E would, after installation, appear in the registry as 0x0006f05e (454750).

Note To enter a value in decimal format, select the **DWORD Decimal** radio button and type in the value data as an integer. This will produce the same formatted value in the registry as entering a DWORD Hex value. For example, a decimal entry of 454750 would also appear as 0x0006f05e (454750) in the registry.

For more information on the registry, refer to Chapter 6, "The Registry."

Running the Installation

Why am I getting a Setup Initialization Error when I run SETUP.EXE?

The error message "Setup requires _SETUP.DLL and _ISRES.DLL (located in _SETUP.LIB) in order to operate properly" may be occurring because the SETUP.EXE file you are trying to run is the copy located in your main Express directory. The file you need to execute will be under the following path (assuming you have specified 1.44 MB disks) is
\\ <AppName>\\ 144MB\\ Disk1\\ SETUP.EXE.

When the Disk Builder creates the installation, it copies SETUP.EXE from the main Express directory into the appropriate installation directory for your application. The path for this directory is created under the location of your setup project file. When you want to execute a setup created by Express, you will ALWAYS launch the SETUP.EXE located in the Disk1 directory. Or, you can launch it simply by clicking the **Test Run** push button on the Setup Checklist.

This message is also displayed when the file _SETUP.LIB (a compressed library containing setup data files) has not been copied by Express to a temporary directory on the target system. The most common reason for this occurrence is that the target system's TEMP variable is set to a directory which either is inaccessible or contains insufficient space for the setup files to be copied and decompressed. Please check that your target machine's TEMP directory is not read-only or located on an unavailable drive. If it contains too many files, try moving or deleting a number of them, and then rerun the installation.

Why isn't my installation waiting for my extension to finish executing?

There are three common reasons for this occurrence.

1. You have unchecked the box marked **Wait for the program to exit before returning to installation** within the Express Extensions **Settings** tab. To correct this, open the Express Extensions dialog box, highlight your executable file in the Extensions window, click on the **Settings** tab, and check the **Wait for the program to exit before returning to installation** box.
2. The Extensions executable has no main window, and therefore has no window handle. When an executable is launched as an extension, Express looks for the window handle of the application, waits in a loop until the handle disappears, and then continues with the installation. If the application doesn't have a window handle, Express will infer that the application has finished executing as soon as it is launched, and so will immediately continue with the setup.
3. The executable launches another application, and then exits while the second application is still running. Express has no knowledge of this second application, so it continues the installation without waiting for the second executable to finish, creating the appearance that it is not waiting for the extension to finish.

Self-registration of .DLLs and .OCXs

How can I use Express to register my .DLL and .OCX files automatically?

InstallShield Express automatically registers any file containing the "OLESelfRegister" string in the version resource.

You can check to see if a third-party .OCX or .DLL file contains this resource by opening the .DLL or .OCX using Borland Resource Workshop or Microsoft Visual C++ and then checking the 'Version Information' resource for the following value:

VALUE "OLESelfRegister", ""

To include this resource in your own .DLLs, create an extra string value field in the version resource for the .OCX/.DLL. This value should have the name OLESelfRegister. The data for this value is ignored and can be set to "".

Microsoft Visual C++ does not currently allow you to add a new value field to the 'Version Information' resource. If you are using Microsoft Visual C++, you will need to add this value directly to the .RC file for your project. You can use any convenient text editor to edit this file and add OLESelfRegister to the existing values in the .RC file. Make sure you rebuild your .DLL/.OCX after making this change.

After you add this string, the version information block will look something like this:

```
VS_VERSION_INFO VERSIONINFO
    FILEVERSION 1,0,0,1
    PRODUCTVERSION 1,0,0,1
    FILEFLAGSMASK 0x3fL
    #ifdef _DEBUG
        FILEFLAGS 0x1L
    #else
        FILEFLAGS 0x0L
    #endif
    FILEOS 0x40004L
    FILETYPE 0x2L
    FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
```

```

BEGIN
  BLOCK "040904b0"
  BEGIN
    VALUE "CompanyName", "InstallShield\ 0"
    VALUE "FileDescription", "RESDLL\ 0"
    VALUE "FileVersion", "1, 0, 0, 1\ 0"
    VALUE "InternalName", "RESDLL\ 0"
    VALUE "LegalCopyright", "Copyright _ 1996\ 0"
    VALUE "OriginalFilename", "RESDLL.dll\ 0"
    VALUE "ProductName", "InstallShield RESDLL\ 0"
    VALUE "ProductVersion", "1, 0, 0, 1\ 0"
    VALUE "OLESelfRegister", ""
  END
END
BLOCK "VarFileInfo"
BEGIN
  VALUE "Translation", 0x409, 1200
END
END

```

You can also manually add the following section to your SWDEPEND.INI file for the installation:

```

[DLLNAME.DLL]
Register=$(DLLSelfRegister)

```

If your SWDEPEND.INI file already contains a section named after your .DLL, you must add the "Register=\$(DLLSelfRegister)" line to the existing section.

Setup Groups and Components

How I do organize my application's files into subdirectories?

In order to arrange your application's files into subdirectories which will be created on the target system, a separate file group must be added for each subdirectory that you wish to include.

In the **Groups and Files** dialog, type the name you want to give your file group in the **Group Name** field. Next, in the **Destination Directory** field, enter the directory in which you would like to install the files that will be included in this group. Do this by selecting one of the Express directory

specifiers available in the drop-down list, and then appending this specifier with a subdirectory, if applicable.

For example, if you have created a group named "Sample Files," and want the files in this group to be installed under a subdirectory of your main application directory, enter `<INSTALLDIR>\SAMPLES` as the group's destination directory. If you then want to add another group of files, named "Advanced Samples," and would like these files installed into a subdirectory of the SAMPLES directory, you would enter

`<INSTALLDIR>\SAMPLES\ADVANCED` as this group's destination directory.

Note You can also locate file groups relative to other group names using the [] delimiters around the group name in the **Destination Directory** field. An entry of "[Sample Files]\ ADVANCED" would then be equivalent to "<INSTALLDIR>\ SAMPLES\ ADVANCED" in the above example.

For more information on Express directory specifiers, please refer to Chapter 3, "Building Your Installation."

How do I add a program group in the Groups and Files dialog?

To add a new group, type the name you want to give the group in the **Group Name** field. (To do this, you must essentially "overwrite" the group name that is currently displayed in this field.) Next, in the **Destination Directory** field, you must specify the directory into which you want to install the group. Once the group name and destination directory have been entered, click the **Add Group** push button.

Note Only after you have entered a unique group name will the **Add Group** push button be enabled. Likewise, only after you have edited an existing group name or destination directory will the **Modify Group** push button be enabled.

For more information, please refer to Chapter 3, "Building Your Installation."

Uninstallation

Why isn't my application appearing in the Add/Remove Programs applet?

There are three possible reasons why your application is not appearing in the Add/Remove Programs applet:

1. Make sure that you have not unchecked the **Automatic Uninstaller** option in the Express **Features** dialog. If you have, please re-check this option and then rerun Disk Builder for your project.
2. Check that you have not left any of the following fields blank in the Application Information dialog: **Application Name**, **Version**, and **Company**. These three values are used as paths for the registry entries that Express automatically makes for your setup. If one or more of these paths is missing, the registry entries required for your setup to have Windows 95 Uninstall capability cannot be made, and therefore your application's name will not appear in the Control Panel's Add/Remove Programs applet.

Please make an entry in each of these three fields, and then rerun Disk Builder. If your main application has no version number, please accept the default of 1.0. If you don't have a company name, just provide a name that you would consider to be a logical substitute for this entry.

3. Make sure that it is a 32-bit installation. A 16-bit installation will not appear in the Control Panel's Add/Remove Programs applet. It will only receive an uninstall icon.

For more information on uninstall functionality, please refer to Chapter 3, "Building Your Installation."

What happens to the uninstall icon if two applications are installed to the same program group?

This depends on whether you're targeting Windows 95 or Windows 3.1.

In Windows 3.1, you can only have one UNINST.EXE icon in a program group. Therefore, if you install a second application's icon(s) to an existing program group, the new application's UNINST.EXE icon will overwrite the

existing UNINST.EXE icon. This will result in the first application being uninstallable.

If you need to avoid this possibility, you can set a unique default program folder name and not include the **Select Program Folder** dialog to appear in your setup. This will ensure that your application's icons would be installed to a unique program group, and also that your customer could not select an existing group, resulting in an existing UNINST.EXE icon being overwritten.

You can also deselect the **Automatic Uninstaller** option located in the Features dialog of Express. This way, your application would not include an UNINST.EXE icon that could overwrite an existing one, but, that would remove the uninstallation functionality for your application.

In a Windows 95 system, uninstallation is not adversely affected by having more than one application's icon(s) installed to the same icon folder. Each application has it's own uninstall information key added to the system's registry, and as the uninstallation of an application is handled through the Windows 95 Control Panel's Add/Remove Programs applet, no UNINST.EXE icon is necessary for any application.

Chapter 8

Express Dialog Boxes

For many users, a chapter consisting solely of the dialog boxes used in the application is a highly valued feature. If you count yourself among this group, you will find this section of the User's Guide very helpful. Every major dialog box that you will use in InstallShield Express Professional is displayed, accompanied by an explanation of what information is to be entered, where the information will be used, and why this information is important.

If you have read through all of the preceding chapters, most of these dialog boxes will be very familiar to you. This chapter will serve as a refresher and a function-specific reference. The dialog boxes are organized in this chapter according to the Setup Checklist heading which they fall under. To find a specific dialog box, either locate its section in the Contents section on this page or look up the dialog box in the Index at the end of the this User's Guide.

Contents

The New Project Dialog Box.....	8-2
Set the Visual Design	8-4
Select InstallShield Objects.....	8-10
Specify Components and Files	8-19
Select User Interface Components.....	8-24
Make System File Changes	8-46
Make Registry Changes.....	8-51
Specify Folders and Icons	8-55
Run Disk Builder	8-59
Test the Installation	8-62
Create Distribution Media.....	8-62

The New Project Dialog Box

This dialog box is unique in that it only appears when you are creating a new setup project file. In many ways, it contains some of the most important project information your installation will need.

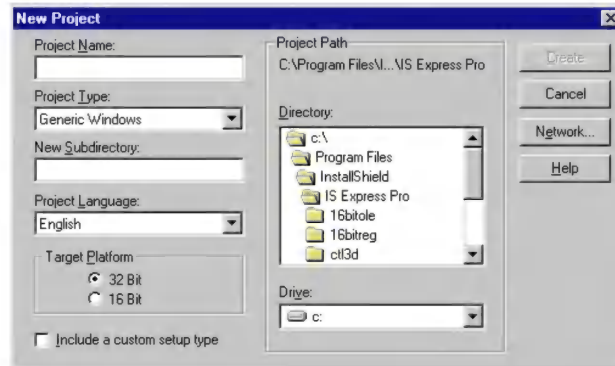


Figure 8.1. New Project dialog box.

The entry fields available here are:

Note All entries made in any of the following fields will be reflected immediately in the Project Path window. This enables you to check the full path name for your setup project file before actually creating the file.

- **Project Name**
Accepts the name to be used as the filename for your application's setup project file. The file extension .IWZ will be added to the entry in this field. All of the settings you create in Express are stored in this .IWZ file. This name is also displayed at the top of the Setup Checklist for this project.
- **Project Type**
Select the development environment that you are using for your application from the drop-down menu. Each project type has only the InstallShield Objects that the selected compiler supports. If the environment you require is not available in the menu, select the **Generic Windows** entry.

- **New Subdirectory**

If you would like a new subdirectory to be added to the Project Path, simply enter its name in this field.

- **Project Language**

Express supports several different languages by simply selecting the desired language from the drop-down menu. For example, to create an installation that displays all of its dialogs and billboards in German, select **German** from the menu. Please check the InstallShield Express website — URL: <http://www.installshield.com/express> — for additional information on new features and functionality, including support for additional languages.

- **Target Platform**

If the development environment selected in the **Project Type** field supports both 16-bit and 32-bit objects, the target platform window will be displayed. Selecting the appropriate radio button for your application ensures that the correct set of InstallShield Objects will be available for your installation.

- **Directory**

Select the directory where this new setup project file is to be located on your system.

- **Drive**

To change the drive structure displayed in the **Directory** window, select a new drive from the drop-down menu. Express displays all available drives on your machine. To map a drive that does not appear in the **Drive** menu, click the **Network...** push button.

In addition, this dialog box has an **Include a custom setup type** checkbox. If you plan on offering a custom setup option to your customers, select this checkbox. For more information on custom setup types, refer to Chapter 3, “Building Your Installation.”

After entering all of the new project information in this dialog box, click the **Create** push button to enter your selections or click the **Cancel** push button to close this dialog without entering the information.

Set the Visual Design

This section of the Setup Checklist deals with entering many of the items that the customer will see on their monitor while your application's installation is running. After entering the necessary information on each of the tabs in the **Set the Visual Design** dialog box, click the **OK** push button to enter your selection and return the Setup Checklist.

App Info Tab

The **App Info** tab of the **Set the Visual Design** dialog box enables you to enter information regarding your application's name and primary executable file.

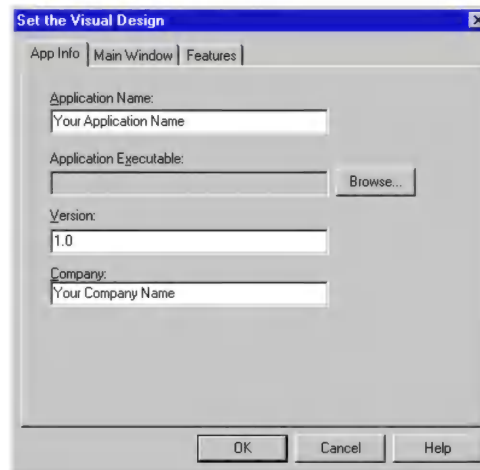


Figure 8.2. The App Info tab of the Set the Visual Design dialog box.

The entry fields available here are:

- **Application Name**
Enter the name of your application in this field. The name that is entered in this field is used in several ways:
 1. It identifies your application in the static text of the Welcome Message and Choose Destination Location user dialogs.

2. It is used as a directory name in the default path in the Choose Destination Location user dialog. The default setting for this dialog is C:\ Program Files\ <Company>\ <AppName>.
3. It is used as a keyname in the paths of two automatic registry entries, the uninstall key (HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows\ CurrentVersion\ Uninstall\ <AppName>) and the registry path which contains the information returned by the User Information user dialog (HKEY_LOCAL_MACHINE\ SOFTWARE\ <Company>\ <AppName>\ <Version>)

For more details about these and other keys created by Express, refer to Chapter 6, "The Registry."

4. It is stored under the value "DisplayName" in the uninstall key shown above. This entry is used to identify your application in the Add/Remove Programs applet.
- **Application Executable**
Enter the file name of your application's primary executable file. This field does not accept a direct text entry. To select your application executable, click the **Browse...** push button to launch the **Open** dialog box. Select the file from this dialog and click the **Open** button to enter the complete path name for the file in the **Application Executable** field. If there are no entries in either the **Groups and Files** dialog box or the **General icon settings** dialog box, Express will copy the executable specified in this field to the Program Files group, and then create an icon entry for the file.

The filename will also be used as the name of the registry key which contains the per-application paths information
HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows\
CurrentVersion\ App Paths\ <EXEC.EXE>

Note If you do not specify an .EXE file in the **Application Executable** field, Express will **not** create the per-application paths entry in the registry.

- **Version**
The default version number for any new Express setup project file is 1.0. If your selected application executable file has an associated version number resource, this will be displayed in the **Version** field. Like the

Application Name field entry, this string is part of the registry key information used for many different functions by Express, including the creation of the uninstall key. Therefore, this field must not be left blank.

- **Company**

Enter the name of your company (80 characters max.). The default setting for this field will be the company name you entered when you first installed Express. The information entered in this field is also used for the user information registry entry. Additionally, like the **Application Name**, it is used as the name of a subdirectory in the **Choose Destination Location** user dialog.

Note It is critical that the **Application Name**, **Version**, and **Company** field are not left blank. Many of the registry keys for your application are built using the information from these three fields. Failure to fill these fields will cause major problems in the installation of your application.

Main Window Tab

The **Main Window** tab allows you to create a distinctive appearance for your setup with minimal effort. Refer to Chapter 2, “A Quick Tutorial” for a step-by-step explanation of how to customize your installation’s display.

Any files selected in this tab do not need to be assigned to setup groups if you are using custom setup types. Express automatically compresses these files in _SETUP.LIB. Express will accept only 16-color bitmaps.

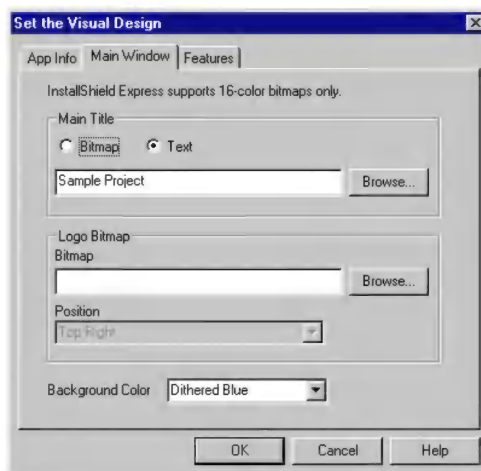


Figure 8.3. The Main Windows tab of the Set the Visual Design dialog box.

This tab is broken down into two windows and one combo box:

- **Main Title**
Select either the **Bitmap** or **Text** radio button, depending on whether or not you have a bitmap which you want to use as the main title displayed on the background of your installation. If you do not want to display a title at top left, clear the edit field.
- **Logo Bitmap**
Use this group box if you have a bitmap (other than a title bitmap) which you want to display in the installation. Click on the down arrow button in the **Position** combo box to select a location for your logo bitmap.

Note If you select Top Left for your logo bitmap, and you also entered text or a bitmap in the Main Title window, the logo bitmap will be displayed behind the main title text or bitmap.

- **Background Color**
Click on the down arrow push button to open the drop-down list which contains the different colors you can select for the background of your setup. Your choices include blue, green, magenta, red, and yellow, all of which can be either dithered or solid.

Features Tab

The **Features** tab contains the **Automatic Uninstaller** check box and the **Target Platform** window.

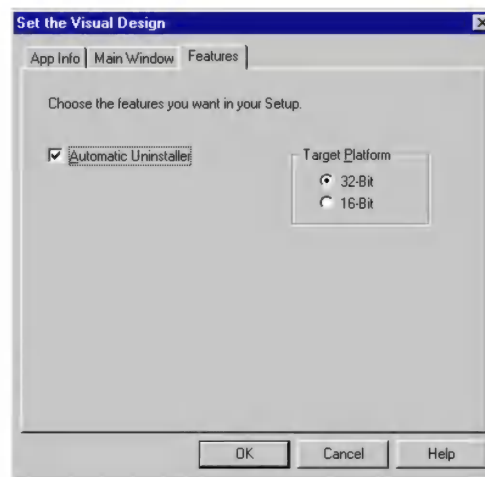


Figure 8.4. The Features tab of the Set the Visual design dialog box.

- **Automatic Uninstaller**

We strongly recommend that you leave the **Automatic Uninstaller** check box in its default position, selected. When this box is checked, Express will automatically include unInstallShield, the InstallShield uninstallation program, with your application. Windows 95 Logo Requirements include uninstallation capability.

If your customer chooses to uninstall your application, unInstallShield can automatically remove your application files, icons, folders, directories, and registry entries. Refer to Chapter 6, “The Registry” for more information.

Note unInstallShield will automatically remove those registry entries which are, by default, created by Express. It will not remove any additional registry keys you may have specified in the **Make Registry Changes** dialog boxes unless you select the **Uninstall Key** check box. No additional uninstall key is required.

- **Target Platform**

The Target Platform window shows you whether you have selected a 16-bit or 32-bit setup when you created this setup project file, and allows you to change the platform if you so desire. If you are creating only one setup to be used for both 16-bit and 32-bit platforms, select the 16-bit radio button.

Note If the platform type for a setup project is changed after any InstallShield Objects have been selected, all existing InstallShield Object files and settings that have been created up to that point will be removed.

Select InstallShield Objects

This section contains all of the dialog boxes presented in the InstallShield Objects section of the Setup Checklist. While you are working on a setup project file, you will only see the dialog boxes applicable to your project type. For this reason, this section is organized by development environment.

Note If you selected **Generic Windows** as the project type, the **Select InstallShield Objects for Visual Basic** dialog box appears.

Specify InstallShield Objects for Borland C++

The **Specify InstallShield Objects for Borland C++** dialog box is comprised of two tabs: General and Advanced.

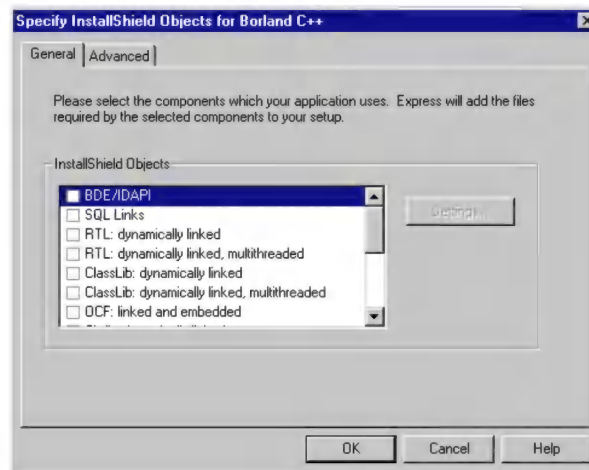


Figure 8.5. The General tab of the Borland C++ Objects dialog box.

To add a component from the list, simply select the check box for the item. Express will automatically create the group(s) and add the files necessary for the InstallShield Object(s) you select. If the selected component has additional customization options, the **Settings...** button will be available. For more information on how to customize an InstallShield Object, refer to Chapter 4, "InstallShield Objects."

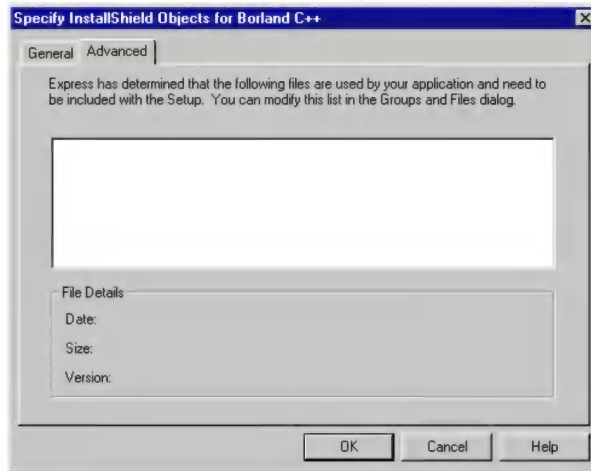


Figure 8.6. The Advanced tab of the Borland C++ Objects dialog box.

The **Advanced** tab of the **Specify InstallShield Objects for Borland C++** dialog allows you to view the file details (date, size, and version, if applicable) of the Borland C++ files which Express has selected to add to your setup. You can reveal or hide the files in the group(s) by clicking on the +/- icon next to the group name. To view the details of a particular file, click on the appropriate filename in the list box.

The **Advanced** tab is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. You should not change the default directories for any of the automatically generated groups. If you need to delete files from any of the automatically generated groups, you can do so in the **Groups and Files** dialog box.

Select InstallShield Objects for Delphi

The **Specify InstallShield Objects for Delphi** dialog box is comprised of two tabs: General and Advanced.

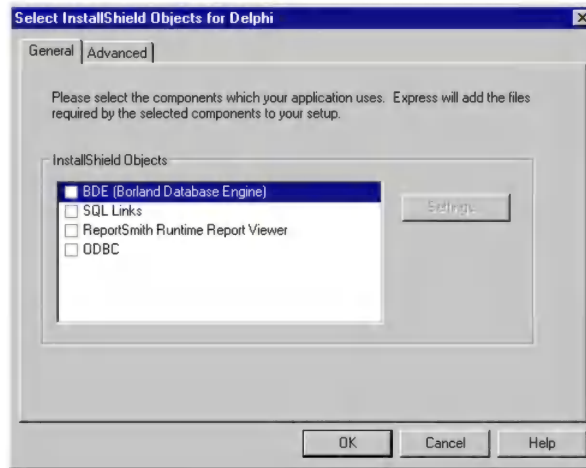


Figure 8.7. The General tab of the Delphi Objects dialog box.

To add a component from the list, simply select the check box for the item. Express will automatically create the group(s) and add the files necessary for the InstallShield Object(s) you select. If the selected component has additional customization options, the **Settings...** button will be available. For more information on how to customize an InstallShield Object, refer to Chapter 4, "InstallShield Objects."

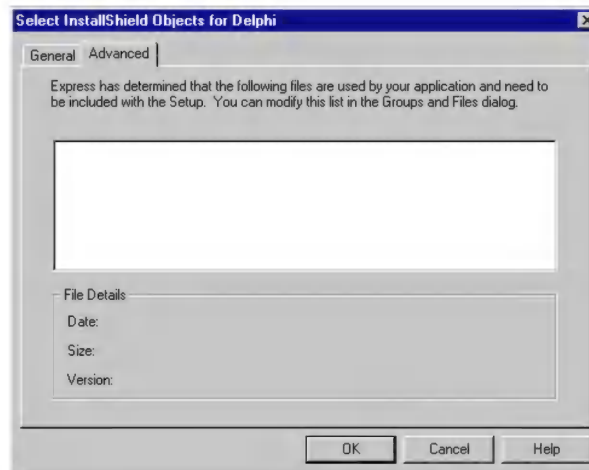


Figure 8.8. The Advanced tab of the Delphi Objects dialog box.

The **Advanced** tab of the **Specify InstallShield Objects for Delphi** dialog allows you to view the file details (date, size, and version, if applicable) of the Delphi files which Express has selected to add to your setup. You can reveal or hide the files in the group(s) by clicking on the +/- icon next to the group name. To view the details of a particular file, click on the appropriate filename in the list box.

The **Advanced** tab is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. You should not change the default directories for any of the automatically generated groups. If you need to delete files from any of the automatically generated groups, you can do so in the **Groups and Files** dialog box.

Specify InstallShield Objects for Paradox

The **Specify InstallShield Objects for Paradox** dialog box is comprised of two tabs: General and Advanced.

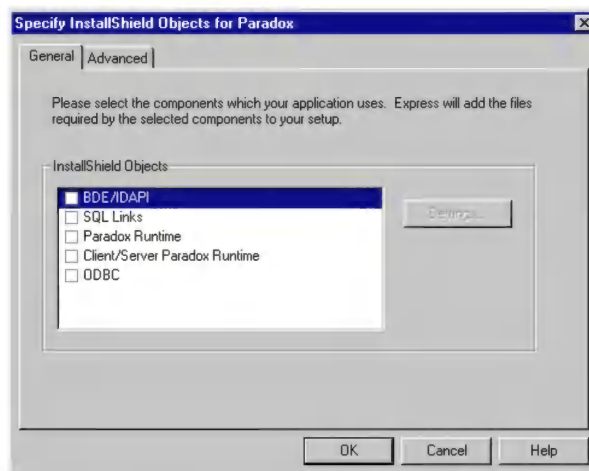


Figure 8.9. The General tab of the Paradox Objects dialog box.

To add a component from the list, simply select the check box for the item. If the selected component has additional customization options, the **Settings...** button will be available. For more information on how to customize an InstallShield Object, refer to Chapter 4, "InstallShield Objects."

You must select either the **Paradox Runtime** or **Client/Server Paradox Runtime** object for Express to include the necessary runtime files for your application. Express will automatically create the group(s) and add the files necessary for the InstallShield Object(s) you select.

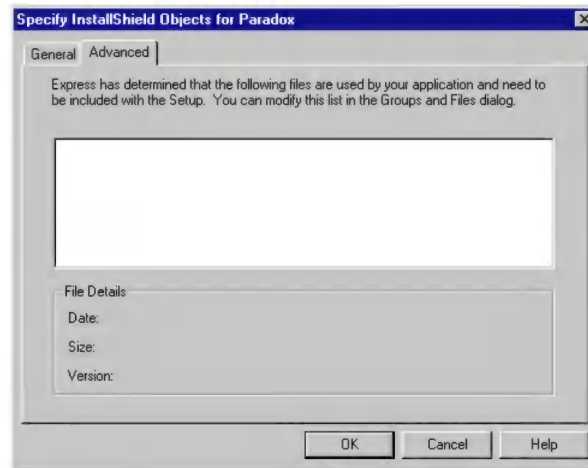


Figure 8.10. The Advanced tab of the Paradox Objects dialog box.

The **Advanced** tab of the **Specify InstallShield Objects for Paradox** dialog allows you to view the file details (date, size, and version, if applicable) of the Paradox files which Express has selected to add to your setup. You can reveal or hide the files in the group(s) by clicking on the +/- icon next to the group name. To view the details of a particular file, click on the appropriate filename in the list box.

The **Advanced** tab is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. You should not change the default directories for any of the automatically generated groups. If you need to delete files from any of the automatically generated groups, you can do so in the **Groups and Files** dialog box.

Select InstallShield Objects for Visual Basic

The **Specify InstallShield Objects for Visual Basic** dialog box is comprised of two tabs: **General** and **Advanced**.

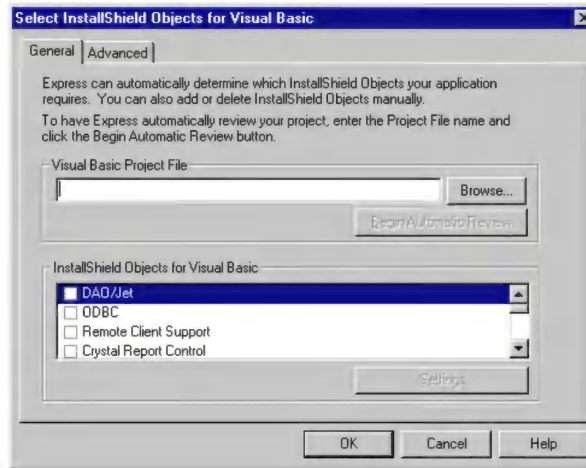


Figure 8.11. The General tab of the Visual Basic Objects dialog box.

The **General** tab is comprised of two windows.

- **Visual Basic Project File**

Express can automatically determine the InstallShield Objects your application requires by checking your application's Visual Basic Project File. To enter the name of your project file, click the **Browse...** push button to launch the **Open** dialog box. Select your Visual Basic project file and click **Open** to close the dialog and enter the filename in the **Visual Basic Project File** field. With a valid file selected, the **Begin Automatic Review** button is available. Clicking this button launches the review process. Express will go through your project file to determine its dependencies. If your project requires any .OCX or other files, Express will add the appropriate files to the System Files - WinSysDir group.

Note Express does not automatically determine dependencies for .VBX files.

- **InstallShield Objects for Visual Basic**

Select the options you want to include with your application. To add a component from the list, simply select the check box for the item. If the selected component has additional customization options, the **Settings...** button will be available. If you select any of the listed controls, Express will add the appropriate files to the System Files - WinSysDir group.

Express automatically sets <WINSYSDIR> as the destination directory for the System Files - WinSysDir group. For more information on how to customize an InstallShield Object, refer to Chapter 4, "InstallShield Objects."

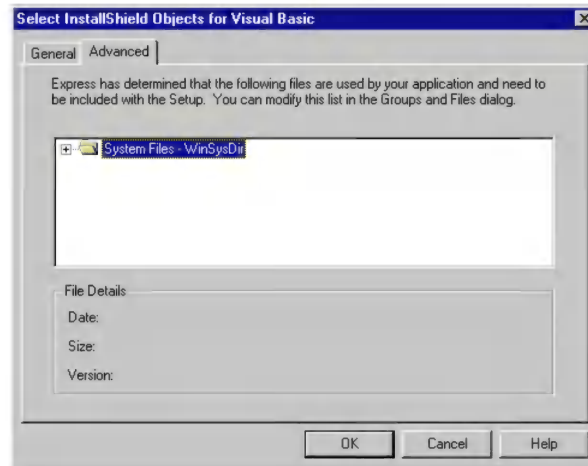


Figure 8.12. The Advanced tab of the Visual Basic Objects dialog box.

The **Advanced** tab of the **Select InstallShield Objects for Visual Basic** dialog allows you to view the file details (date, size, and version, if applicable) of the Visual Basic run-time files which Express has selected to add to your setup. You can reveal or hide the files in the group(s) by clicking on the +/- icon next to the group name. To view the details of a particular file, click on the appropriate filename in the list box.

The **Advanced** tab is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. You should not change the default directories for any of the automatically generated groups. If you need to delete files from any of the automatically generated groups, you can do so in the **Groups and Files** dialog box.

Select InstallShield Objects for Visual C++

The **Select InstallShield Objects for Visual C++** dialog box is comprised of two tabs: General and Advanced.

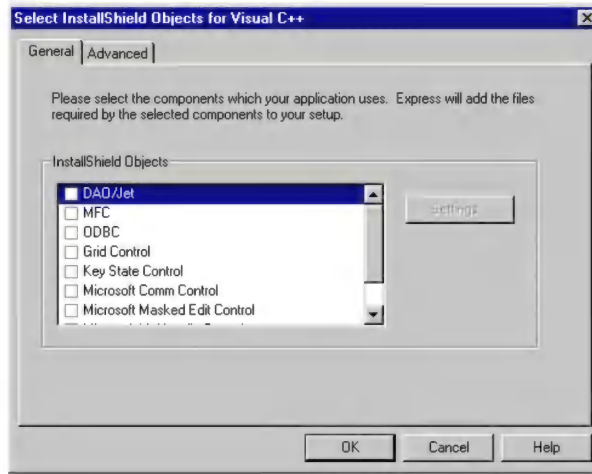


Figure 8.13. The General tab of the Visual C++ Objects dialog box.

To add a component from the list, simply select the check box for the item. Express will automatically create the group(s) and add the files necessary for the InstallShield Object(s) you select. If the selected component has additional customization options, the **Settings...** button will be available. For more information on how to customize an InstallShield Object, refer to Chapter 4, "InstallShield Objects."

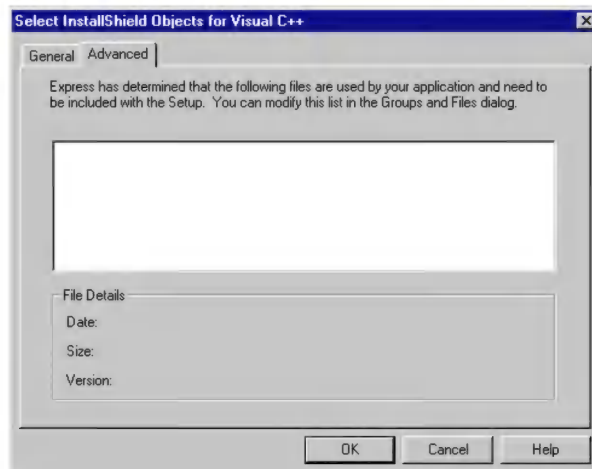


Figure 8.14. The Advanced tab of the Visual C++ Objects dialog box.

The **Advanced** tab of the **Select InstallShield Objects for Visual C++** dialog allows you to view the file details (date, size, and version, if applicable) of the Visual C++ files which Express has selected to add to your setup. You can reveal or hide the files in the group(s) by clicking on the +/- icon next to the group name. To view the details of a particular file, click on the appropriate filename in the list box.

The **Advanced** tab is **not** drag-and-drop enabled. You cannot use it to add or delete files, only to view file information. You should not change the default directories for any of the automatically generated groups. If you need to delete files from any of the automatically generated groups, you can do so in the **Groups and Files** dialog box.

Specify Components and Files

The Specify Components and Files section of the Setup Checklist is where you configure your installation's custom setup options. If you have elected to not offer custom setup type options to your customers, you will work with these dialogs only if your installation requires multiple directories. Express can build your complete setup automatically based on your entries in the other dialog boxes. For an overview of file groups, components, and setup types, refer to Chapter 3, "Building Your Installation."

Groups Tab

The **Groups** tab houses the fields necessary for creating the file groups used to build the custom setup components.

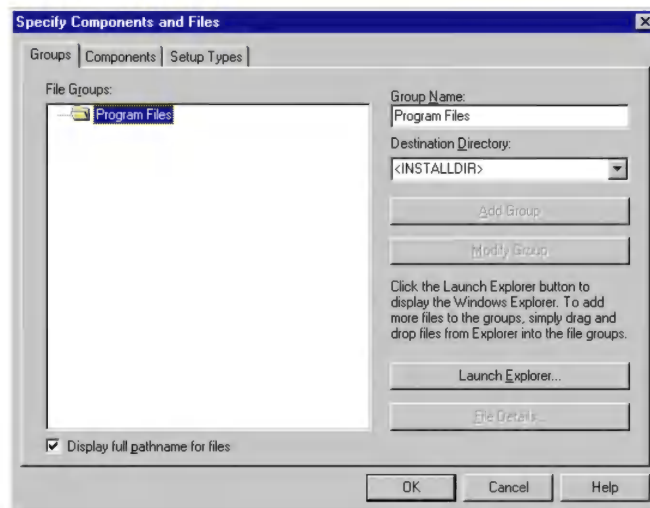


Figure 8.15. The Groups tab of the Specify Components and Files dialog.

The **Groups** tab is comprised of the following fields and windows:

- **File Groups**

The file groups for your setup project file are displayed in this window. If you chose not to offer a custom setup type when you created your new project, the only default group will be Program Files. If you are not offering a custom setup, and if all of your application files can go in the same directory, simply drag and drop them onto the Program Files group.

Note The folder icons in the File Groups window will not reflect the relative location of the group you specified in the **Destination Directory** field. All folder icons are displayed as if they were on the same level.

Once a file has been added to a group, you cannot simply reassign it to another group. If you need to place a file (or files) into a different group, delete it from its current group listing and then add it to the other group. When you add files to a group, InstallShield Express copies the full path and filename of each file. Therefore, do not change the path or filenames of any files after dragging and dropping them onto the groups, or else the Disk Builder will be unable to locate your files when it compiles your setup.

- **Group Name**
Type the name you want to give the group in the **Group Name** edit field. All files assigned to this group will be installed in a single location, the directory listed in the **Destination Directory** field.
- **Destination Directory**
Specify the directory in which you want to install the file group by selecting one of the Express directory specifiers in the drop-down list or typing the desired path.

Your entry in the Destination Directory field is very important, as it will determine the relative location where your files will be copied. Your main installation directory will be the <INSTALLDIR> directory. This specifier will be replaced by the target directory selected by the customer as the destination path during the setup process. Files which must be located in the main Windows directory should be placed in groups with <WINDIR> as the specified destination directory. Files which must be located in the system directory should be placed in groups with <WINSYSDIR> as the specified destination directory.

Note Any files in groups which you are copying to the Windows (<WINDIR>) or System (<WINSYSDIR>) will not be copied if there is a more recent version of the file already in the directory. This feature prevents you from writing over an updated version of a shared file.

You can specify destination directories relative to the locations represented by these specifiers by modifying the particular specifier in the **Destination Directory** field.

- **Display full pathnames for files**
Because the lengths of paths can prevent you from viewing the filenames in the File Groups window, you can deselect the **Display full pathname for files** check box to view the filenames only.

For more information on adding and modifying file groups, refer to Chapter 2, “A Quick Tutorial” and Chapter 3, “Building Your Installation.”

Components Tab

The **Components** tab allows you to create and modify your components, which are the building blocks of a custom setup, and write a description of each to help your customer during the actual installation process.

Note If you have chosen not to offer a custom setup, you do not need to use this dialog. All of your file groups are automatically added to the Application Files component. There is no need to add components, since they are only used to create custom or compact setup types.

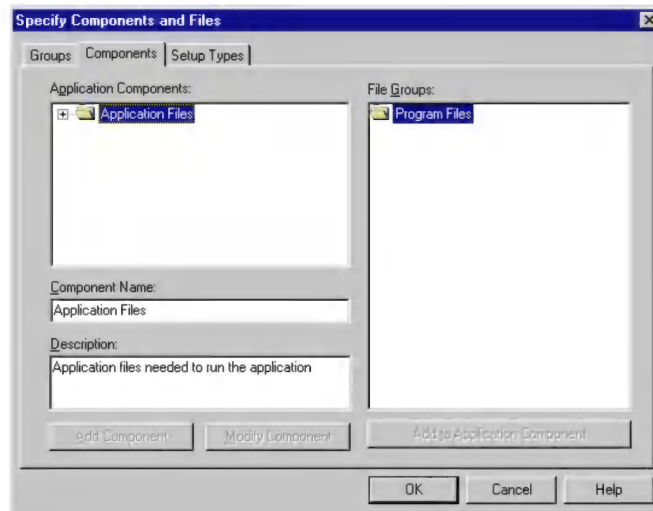


Figure 8.16. The Components tab of the Specify Components and Files dialog.

The **Components** tab is comprised of the following fields and windows:

- **Application Components**
This window displays a list of the current application components. A +/- indicator will be displayed to the left of a component which contains any group(s). You can click on this indicator to hide or reveal the group(s) copied to the particular component.
- **Component Name**
Enter the name you want to give the component in the **Component Name** field.
- **Description**
A well written description should briefly inform the customer about the files and groups in the component, and help him determine whether or not to include the component with the custom installation. This field can contain a maximum of 150 characters. The text you type in this field is displayed in a static text field in the **Custom Setup (Choose Options)** user dialog.
- **File Groups**
Displays a list of the current file groups. Any changes made to a file group must be done in the **Groups** tab of this dialog box.

For more information on adding and modifying components, refer to Chapter 2, "A Quick Tutorial" and Chapter 3, "Building Your Installation."

Setup Types Tab

The **Setup Types** tab allows you to specify the components which will be included with your Custom, Typical and Compact setup types. InstallShield Express only supports these three setup types. You cannot change them or add additional setup types.

Note If you have chosen not to allow custom setup types, you do not need to make any modifications in this dialog. The **Setup Types** dialog will contain only one type, Complete, which will contain your component by default.

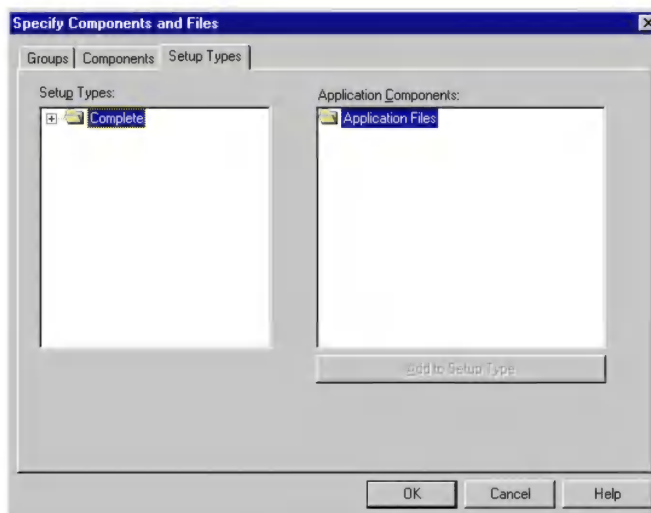


Figure 8.17. The Setup Types tab of the Specify Components and Files dialog.

The **Setup Types** tab consists of the following windows:

- **Setup Types**

This window displays a list of the current setup types. A +/- indicator will be displayed to the left of a setup group which contains any component(s). You can click on this indicator to hide or reveal the component(s) copied to the particular component.

Note The Complete or Custom setup type will always, by default, contain all of your components, including new ones you create. You can delete components from these setup types, but it is not recommended that you do so.

- **Application Components**

Displays a list of the current application components. Any changes made to a component must be done in the **Components** tab of this dialog box.

For more information on modifying setup types, refer to Chapter 2, "A Quick Tutorial" and Chapter 3, "Building Your Installation."

Select User Interface Components

The Select User Interface Components section of the Setup Checklist contains the user dialog boxes and the InstallShield Extensions.

Dialog Boxes

InstallShield Express enables you to customize your application's installation by selecting the dialog boxes that appear while the installation is running.

The following information is true for any of the dialog boxes in the Dialog Boxes section:

- Each of the **Dialog Boxes** dialog boxes consists of the **Settings For** list box, which contains check boxes for selecting and deselecting each of the user interface components in your setup, and a dynamic, tabbed dialog box which displays compressed previews of the user dialogs and allows you to modify the settings for some of them.
- To select or deselect a user dialog, click on the check box next to the dialog name. Note that if you deselect either **Setup Type** or **Custom Setup**, the other is automatically deselected.
- Click the **Preview** push button to see a full-size preview of the highlighted dialog box.
- To modify the settings for a particular user dialog, highlight the dialog and click on the **Settings** tab.

For each of the Dialog Boxes, both tabs will be displayed, if applicable.

Welcome Bitmap

The **Welcome Bitmap** user dialog box allows you to display a bitmap in a child window after the Startup Message, but before any of the other user dialogs.

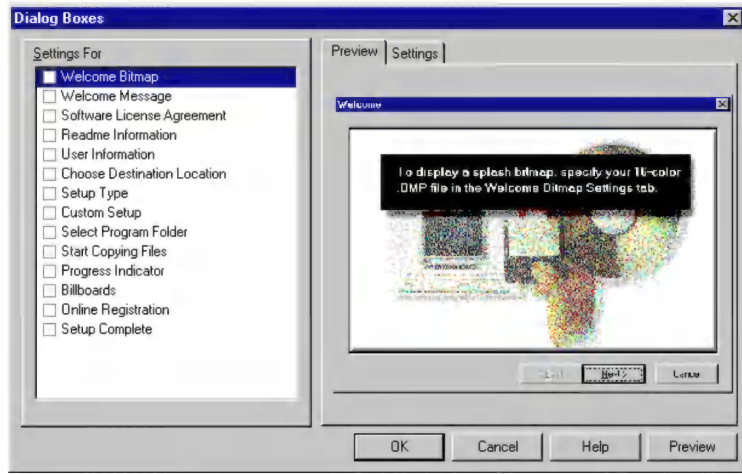


Figure 8.18. The Preview tab of the Welcome Bitmap dialog box.

Note The Welcome Bitmap preview is a sample image built into Express. The previewed image will not change to reflect your specified bitmap.

The **Settings** tab allows you to select the splash bitmap images to be displayed the installation's welcome window.

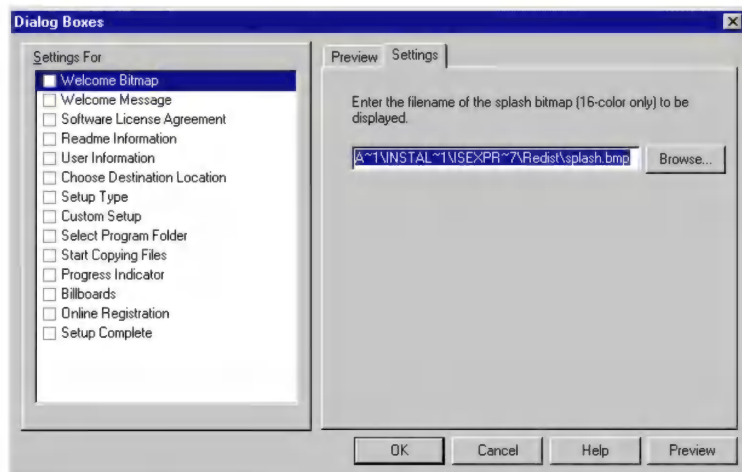


Figure 8.19. The Settings tab of the Welcome Bitmap dialog box.

Specify a bitmap file on the **Settings** tab either by using the **Browse...** push button to search for the file or by typing the fully qualified path and filename in the edit field. The file you select here is compressed into _SETUP.LIB and does not need to be added to a specific file group.

Note Any bitmap that you add cannot exceed 16 color.

If you encounter a “baggage file” error, such as the file being too large, you might need to use a smaller-sized bitmap image. Express will accept images no greater than 444 pixels wide by 275 pixels high.

Welcome Message

The **Preview** tab of the **Welcome Message** user dialog box displays a message welcoming the customer to your application's installation, instructions to exit all Windows programs before starting the setup, and a brief copyright warning.

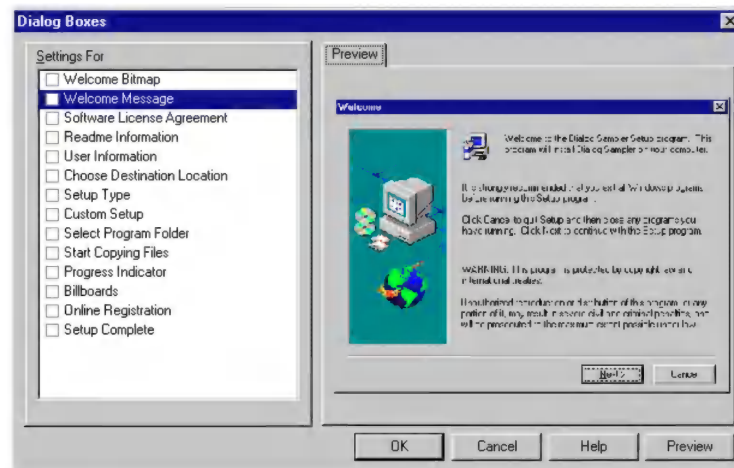


Figure 8.20. The Preview tab of the Welcome Message dialog box.

Express will automatically place the application name you entered in the **Application Information** dialog box in the top static text field of the **Welcome Message** user dialog box. There are no additional settings for this dialog.

Software License Agreement

The Software License Agreement user dialog box allows you to display a text file containing a copy of your license agreement.

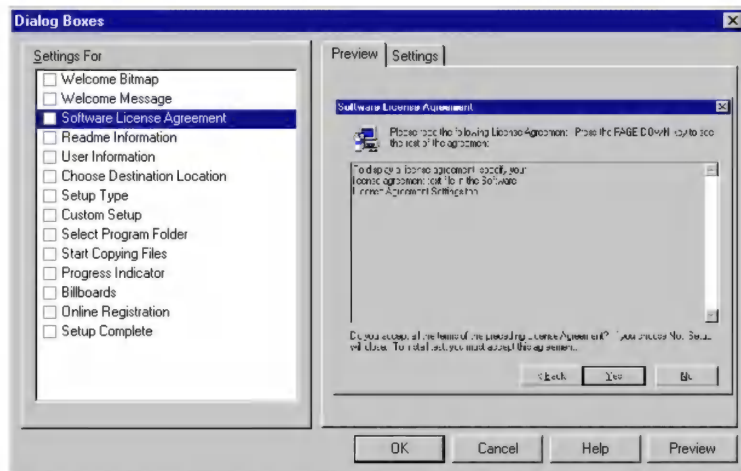


Figure 8.21. The Preview tab of the Software License Agreement dialog.

Note The Software License Agreement preview is a sample image built into Express. The previewed image will not change to reflect your specified text file.

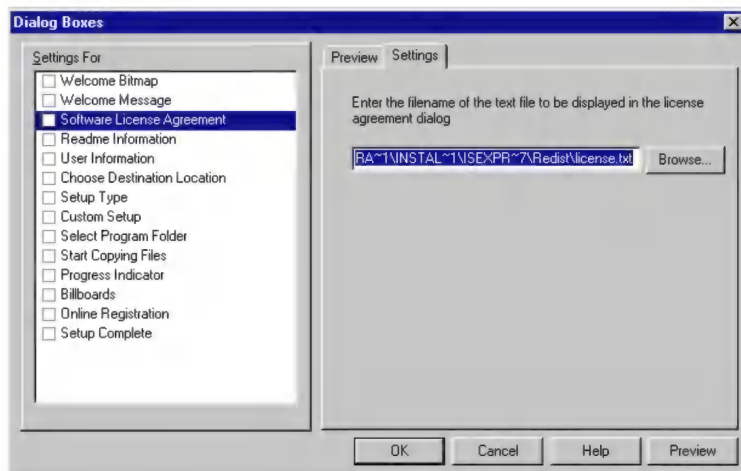


Figure 8.22. The Settings tab of the Software License Agreement dialog.

Specify your file on the **Settings** tab either by using the **Browse...** push button to search for the file on your source system, or by typing the fully qualified path and filename in the edit field. This file is compressed in _SETUP.LIB and does not need to be added to a file group.

Express does not automatically word-wrap your lines of text. If you would like to prevent your customers from having to scroll horizontally through your text file, you will need to manually enter the line breaks in your text.

Readme Information

The **Readme Information** user dialog box allows you to display a Readme text file to the customer before the setup begins collecting information.

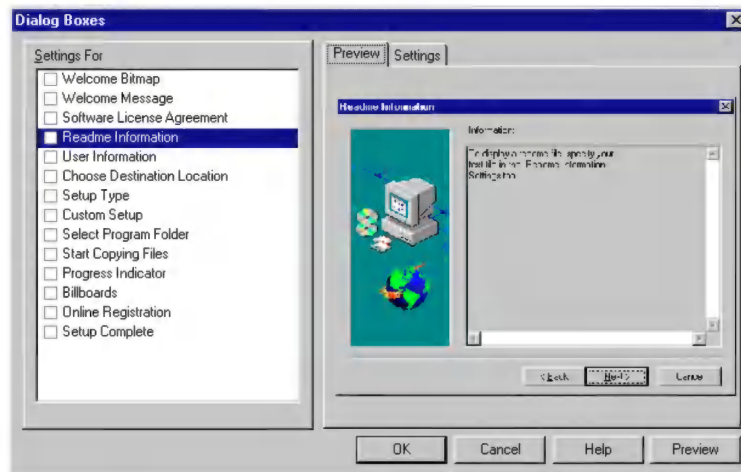


Figure 8.23. The Preview tab of the Readme Information dialog box.

Note The Readme Information preview is a sample image built into Express. The previewed image will not change to reflect your specified text file.

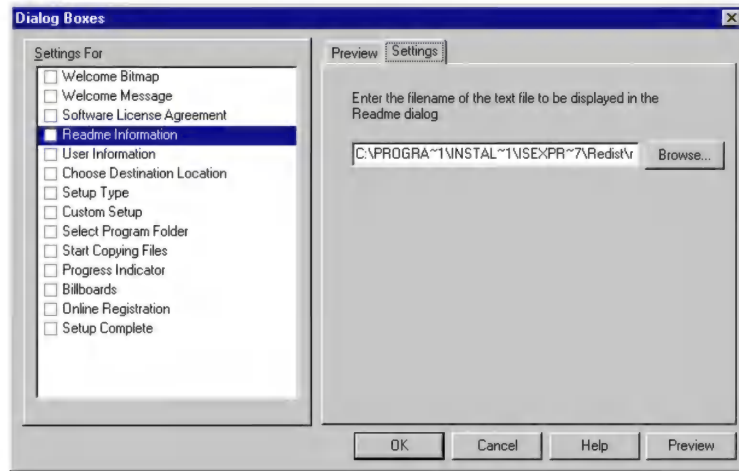


Figure 8.24. The Settings tab of the Readme Information dialog box.

Specify the text file in the **Settings** tab either by using the **Browse...** push button to search for the file on your source system or by typing the fully qualified path and filename in the edit field. This file is compressed in `_SETUP.LIB` and does not need to be added to a file group.

Express does not automatically word-wrap your lines of text. If you would like to prevent your customers from having to scroll horizontally through your text file, you will need to manually enter the line breaks in your text.

User Information

The **User Information** user dialog box collects the customer's name, company, and, optionally, serial number.

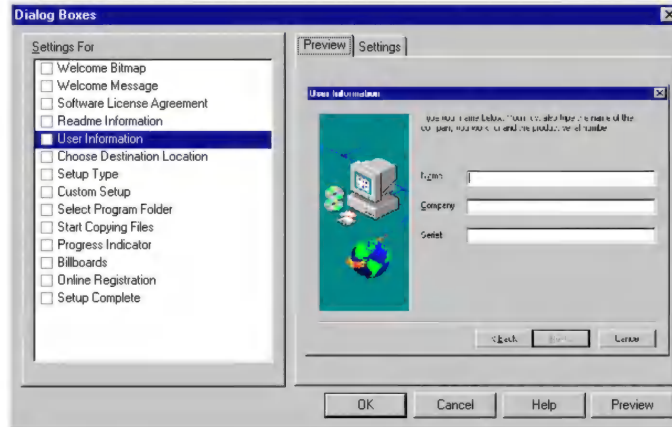


Figure 8.25. The Preview tab of the User Information dialog box.

Note The User Information preview is a sample image built into Express. The previewed image will not reflect your changes.

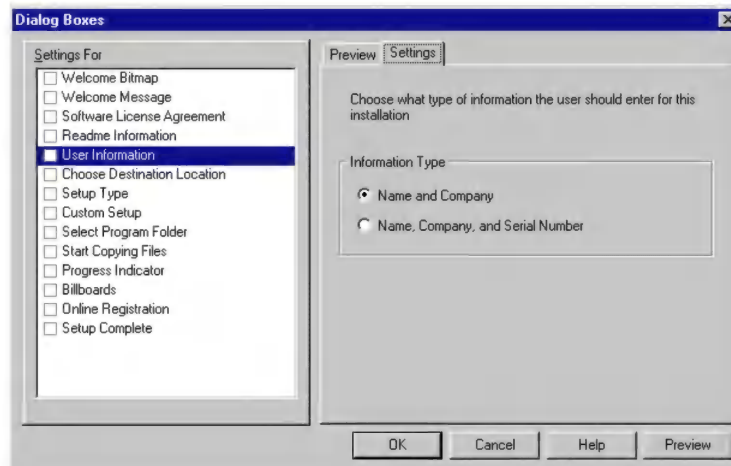


Figure 8.26. The Settings tab of the User Information dialog box.

The default setting for this dialog collects all three pieces of information. If you want to get the customer's Name and Company, but not the serial number, click on the **Name and Company** radio button in the **Settings** tab.

If the target system is Windows 95 or NT:

- The information collected from this dialog will be stored in the registry under the path HKEY_LOCAL_MACHINE\SOFTWARE\ <Company>\ <AppName>\ <Version>
- The key names <Company>, <AppName>, and <Version> will be determined by the information you entered in the **Application Information** dialog box.
- The values collected from the **User Information** dialog box will be stored under the value names Name, Company, and Serial (if applicable).

If the target system is Windows 3.1

- The default entries will be read from the USER.EXE file
- The returned information will be stored in the WIN.INI file. The section will be your application name, and the keywords are Name, Company, and Serial (if applicable).

Choose Destination Location

The **Choose Destination Location** user dialog box allows the customer to select the directory in which he wants to install your application.

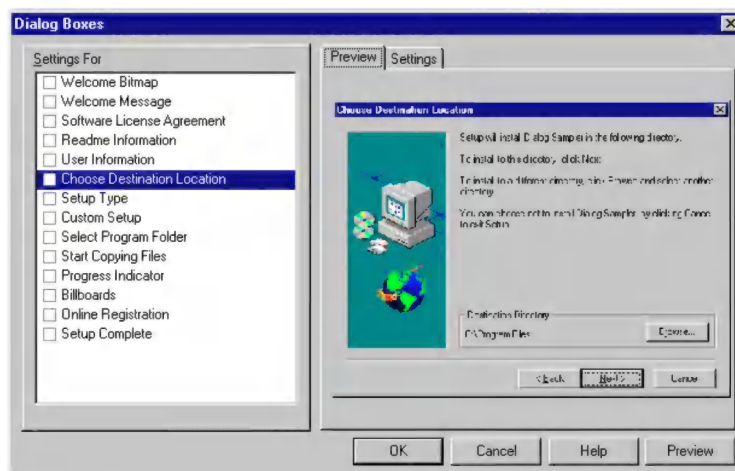


Figure 8.27. The Preview tab of the Choose Destination Location dialog.

The directory and path selected by the customer will be used to replace the <INSTALLDIR> directory specifier in any of the Express dialogs in which you have used it.

Note The Choose Destination Location preview is a sample image built into Express. The previewed image will not reflect your changes.

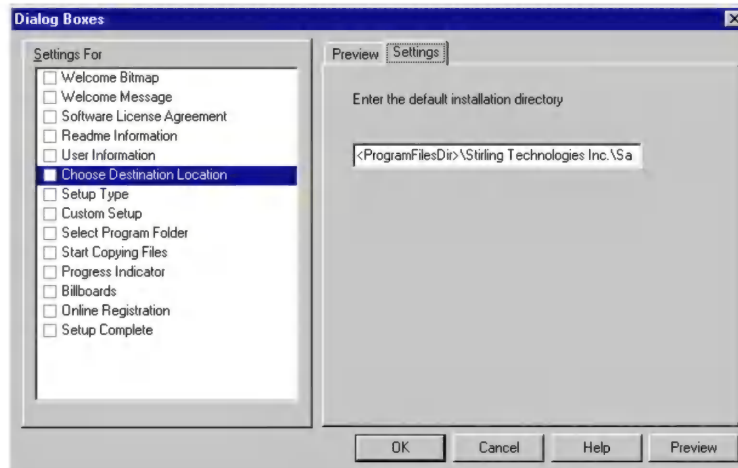


Figure 8.28. The Settings tab of the Choose Destination Location dialog.

You can specify a default directory in the **Settings** tab by typing the path in the edit field. If you are creating a 32-bit application, Microsoft's Windows 95 Setup Guidelines recommend that the destination directory for your application be a subdirectory of the Program Files directory, and that you use a long filename to help create a unique name for it.

In Express, you can designate the Program Files directory on the target system using the <ProgramFilesDir> directory specifier. You can also use the <CommonFilesDir> specifier to indicate the Common Files directory on the target system (e.g. C:\PROGRAM FILES\COMMON FILES).

Note Do not use the <INSTALLDIR> specifier to indicate the default destination directory. The Disk Builder will generate a warning message if you do. Express uses the **Choose Destination Location** user dialog to get the value of <INSTALLDIR> from the customer. Using <INSTALLDIR> in your setting will create an unusable default for your customer.

If you do not specify a directory, Express will set the default to C:\PROGRAM FILES\ <Company>\ <Application>. The names of the <Company> and <Application> directories will be determined by the information entered in the **Application Information** dialog box.

Note If you are creating a 16-bit setup and your setting for this user dialog contains long filenames, Express will automatically truncate the long filenames to the first eight characters, removing any spaces in between characters.

Setup Type

The **Setup Type** user dialog box allows the customer to select a Typical, Compact, or Custom installation. You must create the components for each of these setup types in the **Specify Components and Files** dialog box.

This dialog must be selected if you are offering multiple setup types. If you do not want to give the customer these options, deselect the **Setup Type** user dialog. When you select or deselect the **Setup Type** user dialog, you will automatically select or deselect the **Custom Setup** user dialog, and vice-versa.

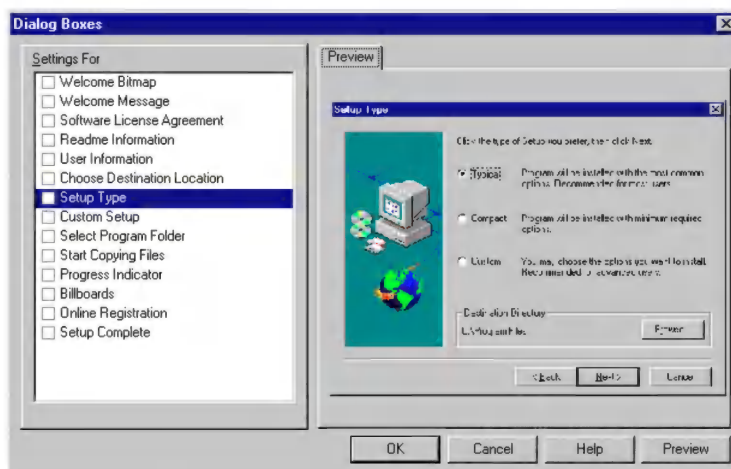


Figure 8.29. The Preview tab of the Setup Type dialog box.

Note The Setup Type preview is a sample image built into Express. The previewed image will not reflect your changes.

The default directory displayed in the **Destination Directory** field is taken from the entry made in the **Choose Destination Location** user dialog box. The customer can select a different destination directory by clicking the **Browse...** push button. Any entry made in this field will override the earlier selections and reset the value of <INSTALLDIR>.

For more information on setup types, refer to Chapter 3, “Building Your Installation.”

Custom Setup

The **Custom Setup** user dialog box will be displayed only if the customer selects a Custom setup type in the **Setup Type** user dialog. The **Select Components** dialog allows the customer to select the components of your application which he or she wishes to install.

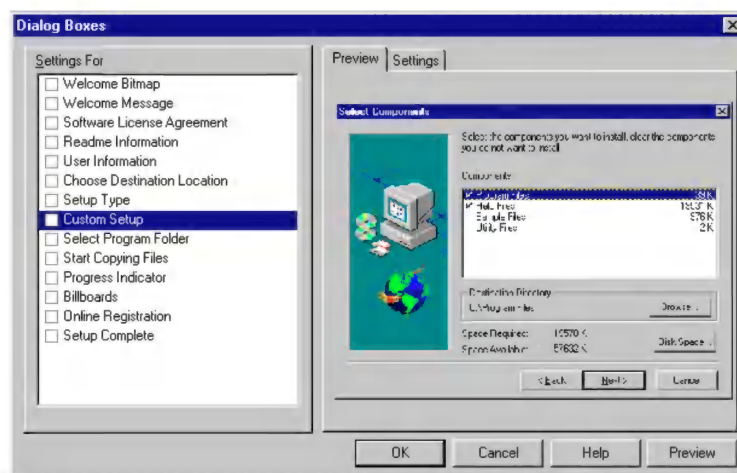


Figure 8.30. The Preview tab of the Custom Setup dialog box.

Note The Custom Setup preview is a sample image built into Express. The previewed image will not reflect your changes.

The default directory displayed in the **Destination Directory** field is taken from the entry made in the **Choose Destination Location** user dialog box. The customer can select a different destination directory by clicking the **Browse...** push button. Any entry made in this field will override the earlier selections and reset the value of <INSTALLDIR>.

For more information on setup types, refer to Chapter 3, “Building Your Installation.”

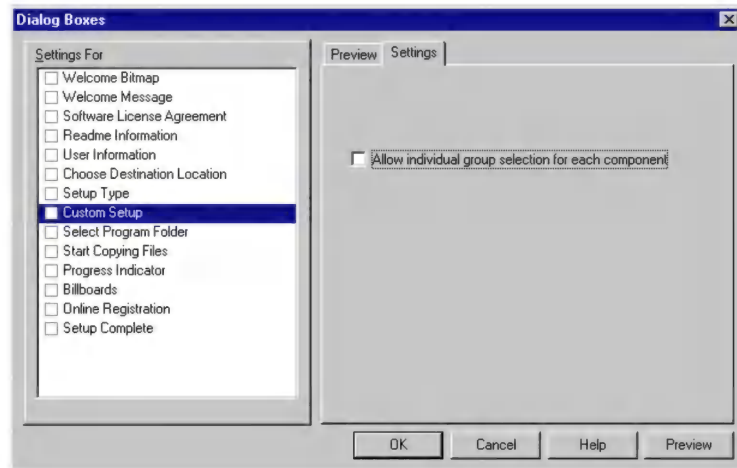


Figure 8.31. The Settings tab of the Custom Setup dialog box.

The **Settings** tab has a check box which lets you determine whether or not the customer will be allowed to select individual file groups. If you select this option, the customer will be allowed to choose which groups, as well as which components, he or she wishes to install.

Select Program Folder

The Select Program Folder user dialog allows the customer to choose the program folder (Windows 95) or group (Windows 3.1 or NT) in which your application icons will be placed.

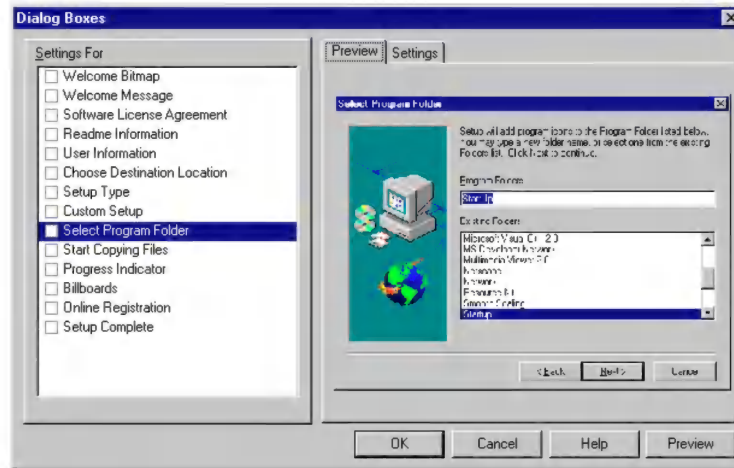


Figure 8.32. The Preview tab of the Select Program Folder dialog box.

Note The Select Program Folder preview is a sample image built into Express. The previewed image will not reflect your changes.

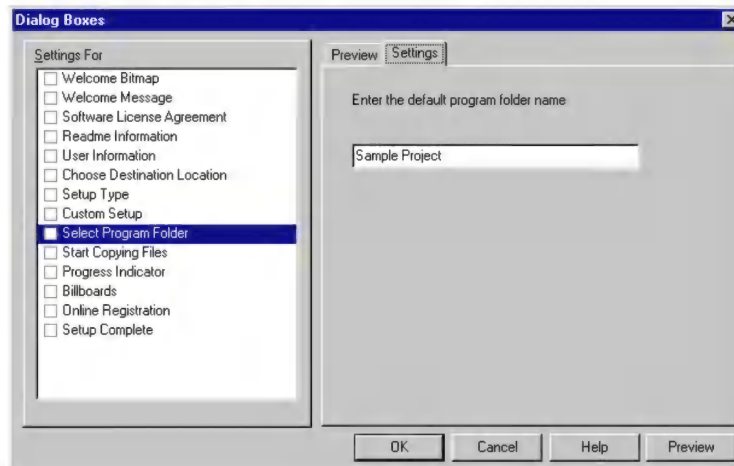


Figure 8.33. The Settings tab of the Select Program Folder dialog box.

Specify the default folder name by typing it into the edit field on the **Settings** tab. If you do not specify a folder name, Express will use your application name as entered in the **App Info** tab as the default folder name.

Start Copying Files

The **Start Copying Files** user dialog box displays the setup type, destination directory, and user information entered by the user. It allows your customer to double-check his entries and, if necessary, go back and modify one or more of them.

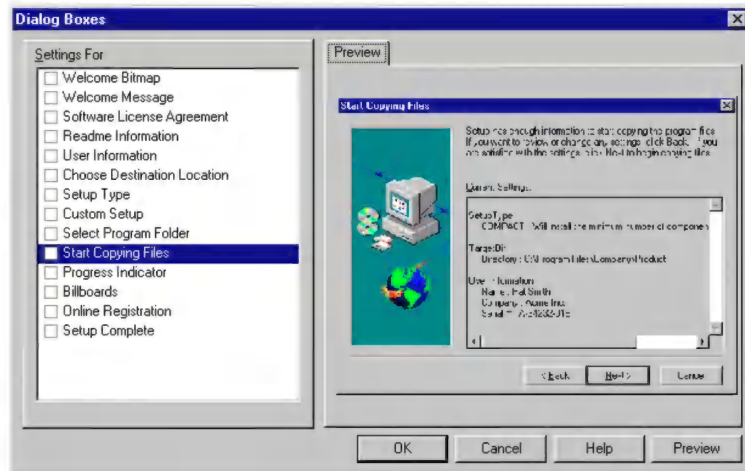


Figure 8.34. The Preview tab of the Start Copying Files dialog box.

Note The Start Copying Files preview is a sample image built into Express. The previewed image will not reflect your application's information.

Since the text displayed in the multi-line edit field of the **Start Copying Files** dialog box is determined by information obtained from the customer, there are no settings for this dialog box.

Progress Indicator

The Progress Indicator gives the customer a graphical representation of the file-transfer process.

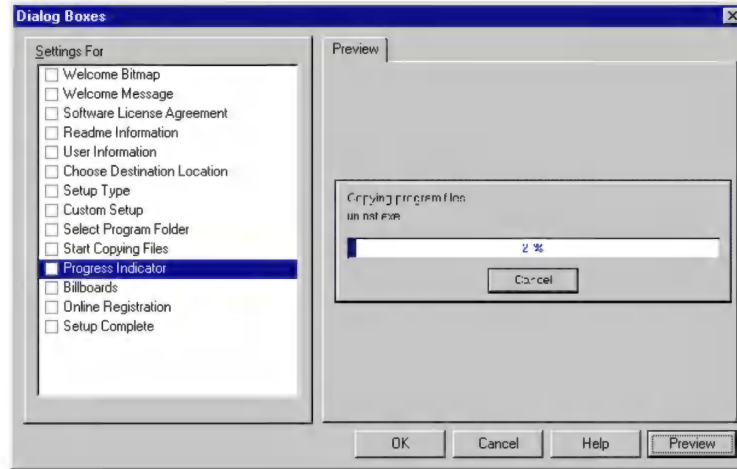


Figure 8.35. The Preview tab of the Progress Indicator dialog box.

The progress bar gradually increments to depict the percentage of the file transfer process which has been completed, while the text field above the bar displays the names of the files as they are copied.

Billboards

Selecting the **Billboards** user interface component allows you to display billboards during your application installation. You can use bitmaps (.BMP) or Windows metafiles (.WMF) as your billboards.

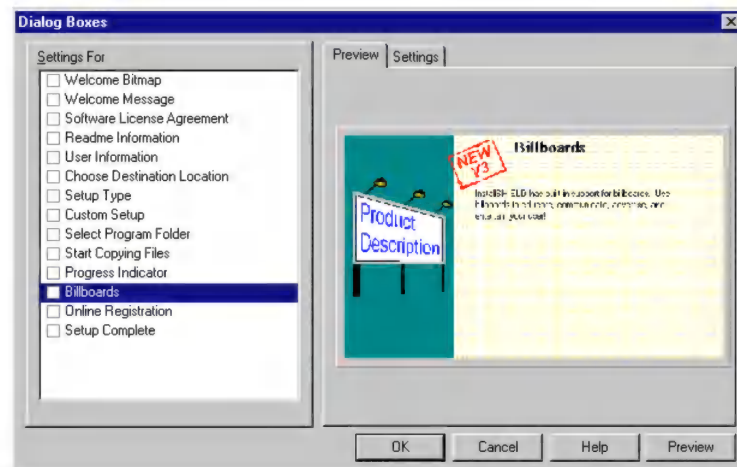


Figure 8.36. The Preview tab of the Billboards component.

Note The Billboards preview is a sample image built into Express. The previewed image will not reflect your application's information.

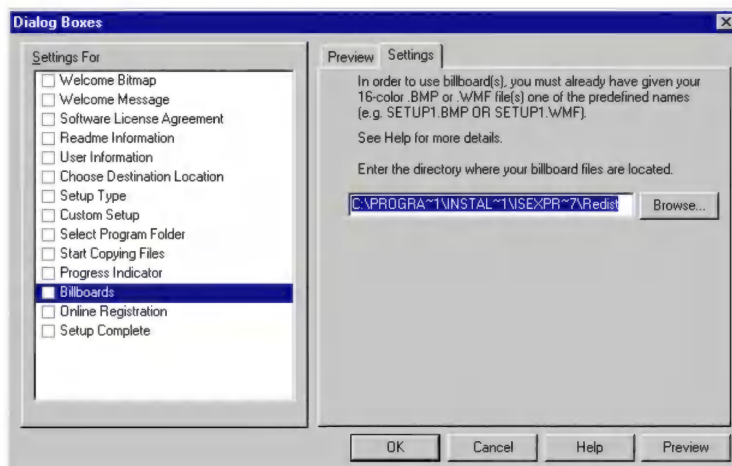


Figure 8.37. The Settings tab of the Billboards component.

In order to use the billboards in Express, you must name your billboard files sequentially (SETUP1.BMP, SETUP2.BMP, etc.) Specify the directory of your file on the **Settings** tab either by using the **Browse...** push button to search for the path or by typing the path in the edit field. InstallShield Express will automatically display the billboard during the file transfer process.

Express includes five sample bitmap billboard files (SETUP1.BMP through SETUP5.BMP) in the REDIST subdirectory of your main Express directory. This directory is also set as the default path for the billboard. You can quickly customize these bitmaps and use them without having to specify a new directory.

Note Any bitmap or metafiles that you include cannot exceed 16 colors.

If you encounter a “baggage file” error, such as the file being too large, you might need to use smaller-sized images or reduce the number of billboards.

Online Registration

The latest software applications include an online registration package to help the customers with modem-enabled computers to quickly register their copy of the product. InstallShield Express includes an Online Registration feature serviced by Pipeline Communications, Inc.

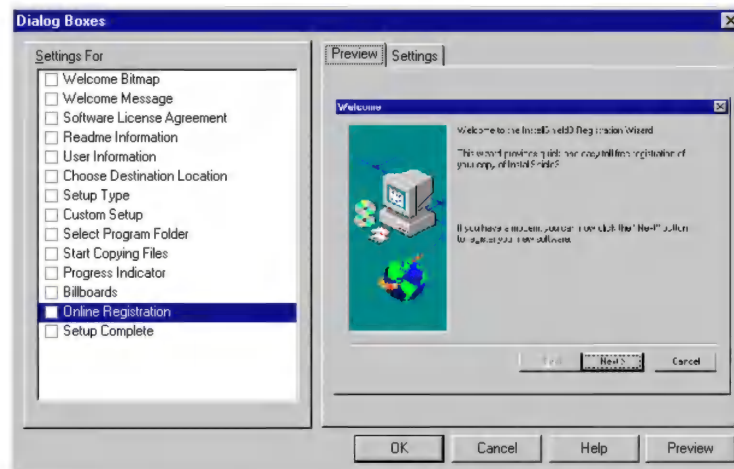


Figure 8.38. The Preview tab of the Online Registration dialog box.

In order to activate this feature, you must contact Pipeline at 1-800-WIN95REG, or by email at WIN95REG@pcpipeline.com.

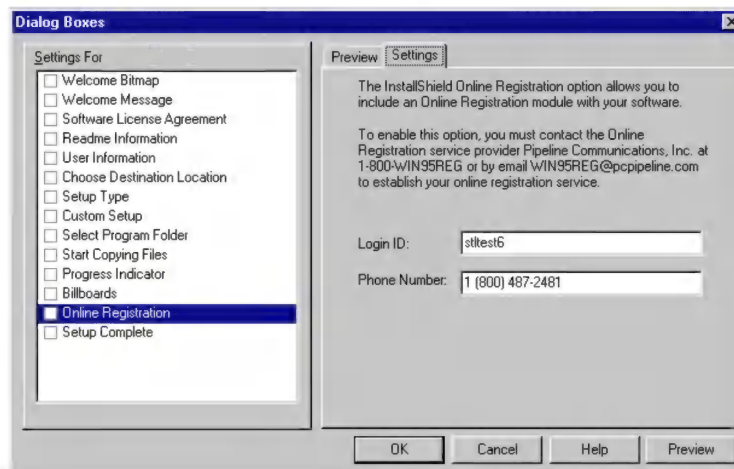


Figure 8.39. The Settings tab of the Online Registration dialog box.

Once you have established service, Pipeline Communications will give you the **Login ID** and **Phone Number** which must be entered in the **Settings** tab.

The online registration will take the user information entered by the customer (name, company, version number, etc.) at the start of the installation and transfer it by modem to the specified site.

Setup Complete

The **Setup Complete** user dialog box actually consists of two dialogs: a reboot computer dialog box and a launch application dialog box. The preview shows only the reboot computer dialog box.

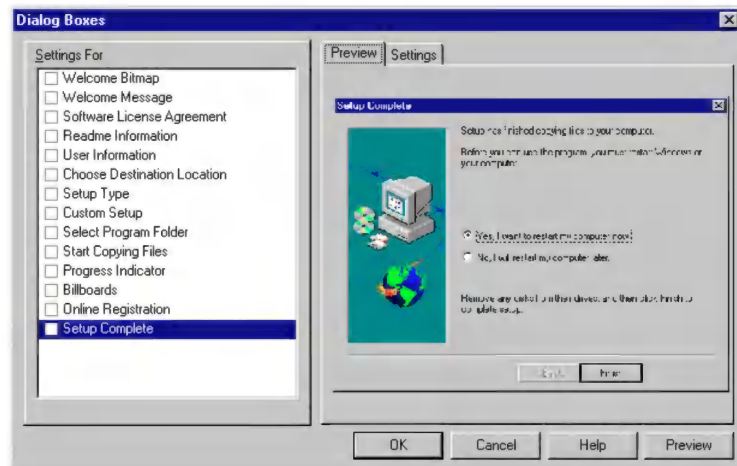


Figure 8.40. The Preview tab of the Setup Complete dialog box.

The reboot computer dialog box informs the customer that the installation is complete and gives him the option to reboot his computer. (In Windows 3.1 only, it will also offer the option to restart Windows). This is, of course, especially useful if your installation has made system changes (such as adding a .VXD) that require a restart.

InstallShield will automatically display the reboot computer dialog if your setup encounters any locked or shared files. You can also display the dialog for all installations by selecting the **Always show reboot computer dialog** check box in the **Settings** tab.

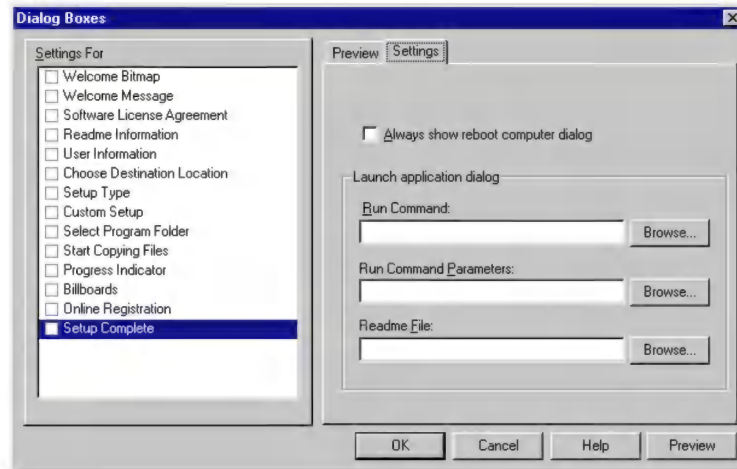


Figure 8.41. The Settings tab of the Setup Complete dialog box.

The Launch application dialog window allows the customer to run your application or view your Readme file by selecting a check box and clicking OK. You can offer the option to launch both the application or Readme, either one separately, or neither.

To offer the launch application option:

- Click the **Browse...** push button next to the **Run Command** field to open the **Setup Files Browser** dialog box. Highlight your primary executable from among the files you have added to your groups in the Groups and Files dialog box. Click **OK** to accept your choice or **Cancel** to close the dialog.

Note You must have either copied files into groups in the **Groups and Files** dialog box or specified an application executable in the **Application Information** dialog box in order to select a file from the Setup Files Browser. If you want to offer the option of launching a Readme file, you must first copy the Readme file to a group, then return to the **Setup Complete** user dialog box and select the file.

- You can also specify your executable by typing the installation path and filename in the **Run Command** field. Because the program will be launched after the installation, the path must be specified using one of the Express directory specifiers.

If you need to add a parameter to your launch command, specify it also by either selecting a file from the Setup Files Browser or typing your command line option(s) in the **Run Command Parameters** field. You can use an Express directory specifier in this field also.

To offer the launch Readme file option:

- Select your Readme file using the Setup Files Browser feature or use your keyboard to make your file selection in the **Readme File** field. (As with the executable file above, use a group name or directory specifier for the path).

Note If you do not select an executable or Readme file in the Launch application dialog window, but the **Setup Complete** reboot computer user dialog is displayed, the launch application dialog box will also be displayed, though without any options for the user.

Express Extensions

InstallShield Extensions allow you to call an external .DLL or launch an .EXE file during your installation.

Note InstallShield Extensions are designed for use by the more experienced developer. If you are uncertain of the implications of using an InstallShield Extension in your installation, please do not attempt using them. For more information on this topic, refer to Chapter 5, "InstallShield Extensions."

If no extensions have been created yet, the **Express Extensions** dialog box will have only a single tab titled **Ordering**.

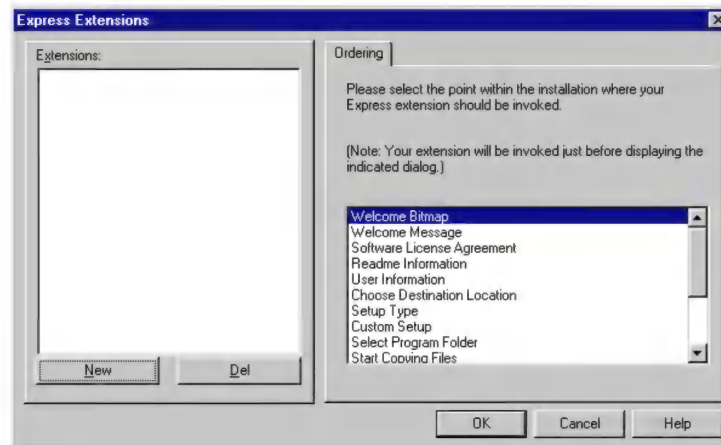


Figure 8.42. The Ordering tab of the Express Extensions dialog box.

The **Ordering** tab consists of the Extensions window and the selected User Interface components.

- **Extensions**

The Extensions window displays all of the current InstallShield Extensions included in the installation. To add an InstallShield Extension, click the **New** push button beneath the Extensions window to open the **New Extension** dialog box. For more information on adding an InstallShield Extension, refer to Chapter 5, "InstallShield Extensions."

Note If you have multiple extensions, the order that they appear in the Extensions window does not reflect the order that they will be executed during the installation.

To delete an extension, highlight the extension you want to remove and click the **Del** push button beneath the Extensions window.

- **Ordering**

After creating your extension, you must select where in the installation process the extension will be called. Highlight one of your Express user dialogs to select the point within your installation when you want your extension to be launched. The extension will run **BEFORE** the dialog highlighted. Only the dialogs that were selected in the **Select User Interface Components Dialog Boxes** appear in the **Ordering** tab.

Note If you have already included an extension and are creating an additional one, do not highlight the location of the new extension until you have added it. If you select a location before clicking the **New** push button, you will change the order of the existing extension which is currently highlighted

Once you have selected the point within the setup at which your extension will run, click the **Settings** tab.

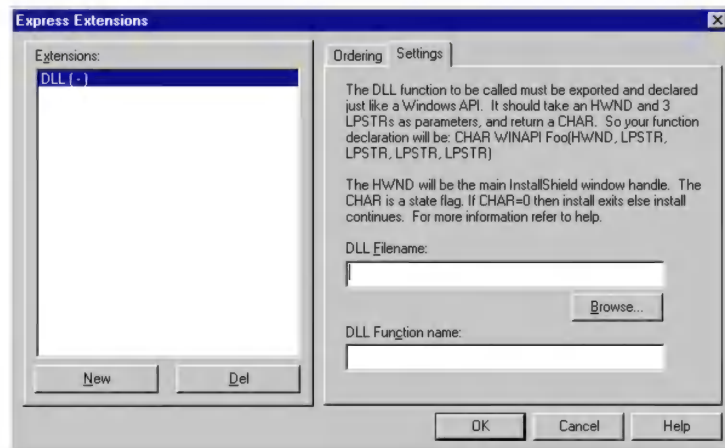


Figure 8.43. The Settings tab of the Express Extensions dialog box.

The Extensions window visible in the **Ordering** tab is still present. The **Settings** tab accepts information specific to the type of extension being added. Enter the settings for the appropriate type of extension:

- **.DLL function**
Use the **Browse...** button to select the .DLL. Type the name of the function you wish to call in the DLL function name field.
- **.EXE file**
Use the **Browse...** button to select the .EXE file. There is also a **Optional Program Parameters** field for entering additional information. Select the **Wait for the program to exit before returning to installation** check box if you want your setup to wait for the .EXE file to finish executing before continuing the installation. Finally, there is a field which allows you to enter disk label text if your .INS file will require the user to insert a separate disk.

Make System File Changes

The **Make System Files Changes** dialog box contains four tabs. These tabs enable your installation to make changes to various files on your customer's system.

Private .INI Files Tab

The **Private .INI Files** tab allows you to make changes to any of your .INI files after the user has specified the necessary setup information.

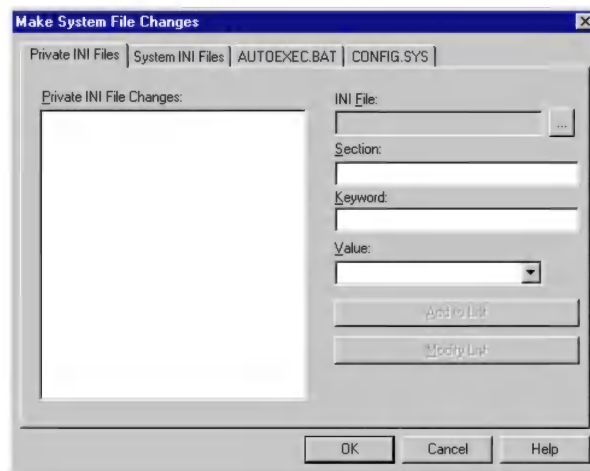


Figure 8.44. The Private .INI Files tab of the Make System File Changes dialog.

The **Private .INI** tab contains the following areas:

- **INI File**
Click on the browse (...) push button to open the Setup Files Browser. Select the file you want to change from among the files which have already been assigned to groups in the **Groups and Files** dialog box. The name of the file you select will be displayed in the **INI File** field.
- **Section**
Enter the name of the section which contains the change in the **Section** field.
- **Keyword**
Enter the appropriate keyword in the **Keyword** field.

- **Value**
Specify the value you want to change or add in the **Value** combo box. In order to take advantage of installation information, you can use Express directory specifiers in this box.
- **Private INI File Changes**
When you finish making your first change to an .INI file, the filename is displayed, along with a +/- indicator and a small folder icon, in the Private .INI File Changes window. You can click on the +/- indicator to hide or reveal your information.

System .INI Files Tab

The **System .INI Files** tab allows you to make changes to the .INI files on the target system after the user has specified the necessary setup information.

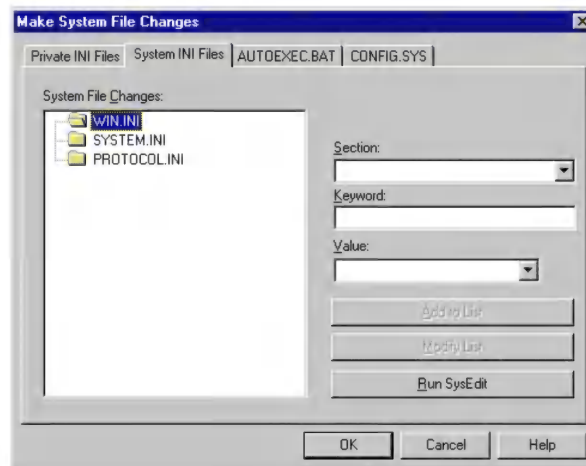


Figure 8.45. The System .INI Files tab of the Make System File Changes dialog.

The **System .INI** tab contains the following areas:

- **Section**
Enter the name of the section which contains the change in the **Section** field.
- **Keyword**
Enter the appropriate keyword in the **Keyword** field.

- **Value**
Specify the value you want to change or add in the **Value** combo box. In order to take advantage of installation information, you can use Express directory specifiers in this box.
- **System File Changes**
When you finish making your first change to an .INI file, the filename is displayed, along with a +/- indicator and a small folder icon, in the System File Changes window. You can click on the +/- indicator to hide or reveal your information.

AUTOEXEC.BAT Tab

The **AUTOEXEC.BAT** tab allows you to make changes to the AUTOEXEC.BAT file on the target system after the customer has specified the necessary setup information.

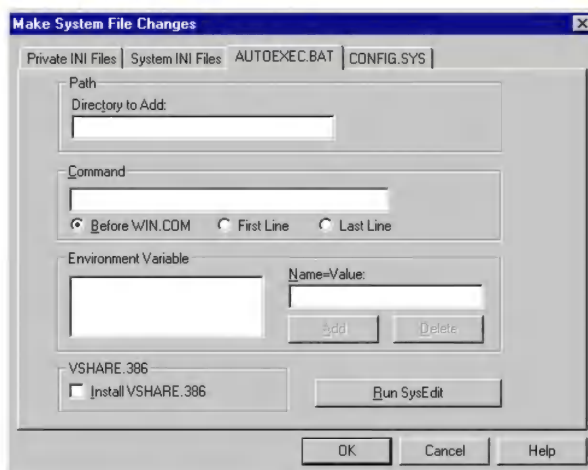


Figure 8.46. The AUTOEXEC.BAT tab of the Make System File Changes dialog.

The **AUTOEXEC.BAT** tab contains the following areas:

- **Directory to Add**
Enter the path in the **Directory to Add** field. In order to take advantage of installation information, you can use Express directory specifiers in your entry.

- **Command**

Enter the command in the **Command** field (128 characters max.). You can use Express directory specifiers in this field.

Click one of the radio buttons beneath the **Command** field to specify whether you want the command to be added before WIN.COM, as the first line of the AUTOEXEC.BAT file, or as the last line of the file.

- **Name=Value**

In the **Name=Value** field, enter the variable name and its value in the proper format (e.g. Wizdata=[Data Files]). You can use Express directory specifiers in this field.

- **Environment Variable**

Your entry in the **Name=Value** field will be displayed in the Environment Variable list box

Note If you create an entry with the same name as an existing environment variable in the target AUTOEXEC.BAT, InstallShield Express will reset the entry on the target system. If you do not want to do this, make sure you use unique names for your environment variables.

- **VSHARE.386**

To install VSHARE.386, click the **Install VSHARE.386** check box.

CONFIG.SYS Tab

The **CONFIG.SYS** tab allows you to make changes to the CONFIG.SYS file on the target system after the customer has specified the necessary setup information.

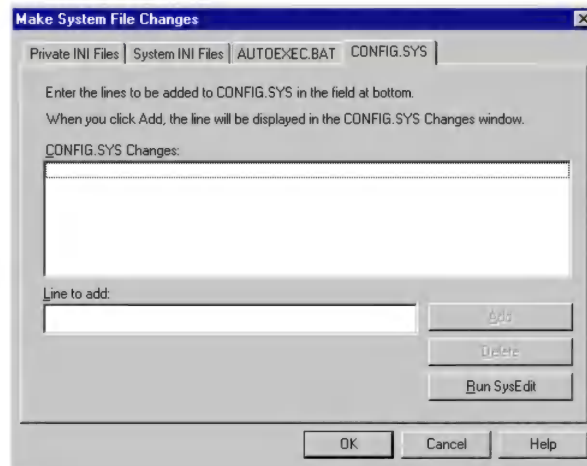


Figure 8.47. The CONFIG.SYS tab of the Make System File Changes dialog.

The **CONFIG.SYS** tab contains the following areas:

- **Line to add**
To add a setting to the CONFIG.SYS file, enter your line in the **Line to add** field (128 characters max.), then click the **Add** push button. In order to take advantage of installation information, you can use Express directory specifiers in your entry.
- **CONFIG.SYS Changes**
After adding a line, it will be displayed in the CONFIG.SYS Changes window.

Make Registry Changes

The Make Registry Changes section enable you to add or modify keys and values in your customer's registry. The **Make Registry Changes** dialog box contains three tabs.

Registry - Keys Tab

The **Registry - Keys** tab allows your installation to add keys for your application to the customer's registry.

Note For additional information on using InstallShield Express to work with the registry, refer to Chapter 6, "The Registry."

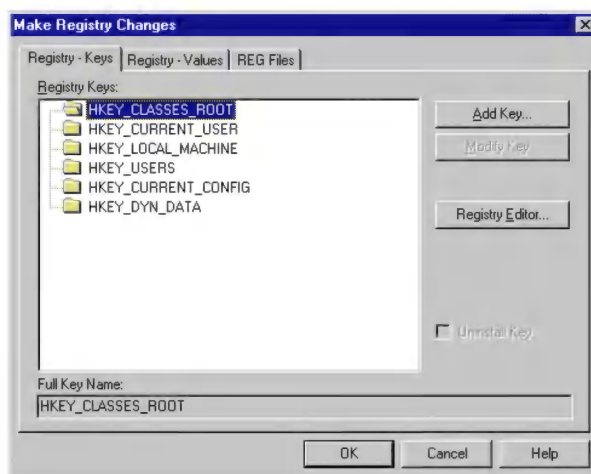


Figure 8.48. The Registry - Keys tab of the Make Registry Changes dialog.

This dialog is dynamically linked with the **Registry - Values** tab, which lets you add or modify the values of the highlighted key in the **Registry - Keys** tab. This tab contains the following areas.

- **Registry Keys**
This window displays all of the current keys to be added for this installation. The six root keys will always be present in the window. When a key has one or more subkeys, a +/- indicator is displayed next to the folder icon. You can click on this indicator to hide or reveal the

subkeys. To add a subkey to one of the root keys, click the **Add Key...** push button to display the **Add Key** dialog box.

Note You can enter more than one level of subkeys at once using a backslash to separate them, as shown below:
Software\ Company\ MyApp\ 3.0\ Help Files

To modify an existing subkey, click the **Modify Key...** push button and make any necessary changes in the **New Key** field.

- **Uninstall Key**

To add the new key to the uninstall log file, select the **Uninstall Key** check box.

- **Full Key Name**

The full path and name of the highlighted key is always displayed in the Full Key Name static text field, truncated at the right.

To change the value for a registry key, highlight the key in the Registry Keys window and select the **Registry - Values** tab.

Registry - Values Tab

The **Registry - Values** tab allows you to set name/data pairs for the key which was highlighted in the Registry Keys window when you clicked on the **Registry - Values** tab.

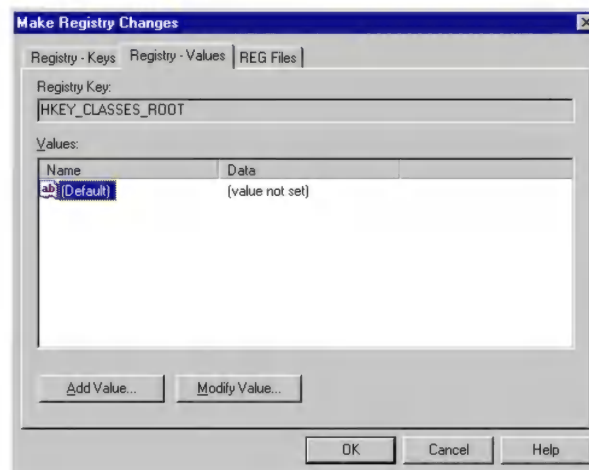


Figure 8.49. The Registry - Values tab of the Make Registry Changes dialog.

The **Registry - Values** tab is comprised of two sections:

- **Registry Key**

The name of the key being modified is displayed in the **Registry Key** static text field. If you want to add to or modify values from a different key, simply click on the **Registry - Keys** tab, highlight the key you want to update, and click the **Registry - Values** tab again.

- **Values**

The Values window displays the current settings for the selected registry key. To add or modify the highlighted value, click the appropriate push button at the bottom of the tab and enter the change in the dialog box. You can use one of the Express directory specifiers in your data.

For additional information on using InstallShield Express to work with the registry, refer to Chapter 6, “The Registry.”

REG Files Tab

The **REG Files** tab allows you to merge a selected .REG file into your customer’s registry.

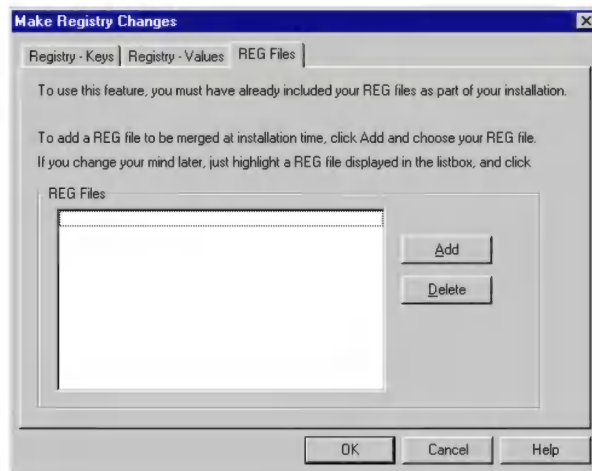


Figure 8.50. The REG Files tab of the Make Registry Changes dialog.

The **REG Files** tab consists of a single window. The REG Files window displays the .REG files to be merged during the installation.

Note To include a .REG file, it must first be added to one of the file groups for your installation. For more information on file groups, refer to Chapter 3, “Building Your Installation.”

To add a .REG file to the REG Files window, click the **Add** push button to open the Setup File Browser.

To delete a .REG file from the REG Files window, highlight the .REG file and click the **Delete** push button.

Specify Folders and Icons

The Specify Folders and Icons section of the Setup Checklist contains a single dialog box with two tabs.

General Settings Tab

The **General** icon settings tab allows you to specify the icons you want to place in your application's folder (Windows 95) or group (Windows 3.1 and NT). On Windows 95 systems, a folder with the Program name will be placed on the Start Programs menu, with all other icons available underneath.

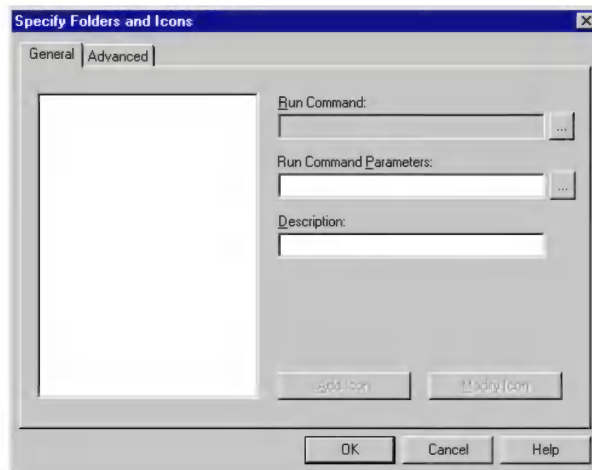


Figure 8.51. The General tab of the Specify Folders and Icons dialog box.

The **General** tab contains these fields:

- **Run Command**
This field contains the installation path of the file containing the icon resource.

To add an icon, click on the browse (...) button next to the **Run Command** field. The **Setup Files Browser** dialog box opens, displaying the files which you have previously copied to your program groups in the **Groups and Files** dialog box. Select the program file to which you want to assign an icon, and click **OK** to close the window.

When you return to the **General** tab, the installation path of the file (e.g. [Program Files]\YOURAPP.EXE) will be displayed in the **Run Command** field.

- **Run Command Parameters**

You can specify a command parameter by either clicking on the browse button next to the box and selecting an installation path from the **Setup Files Browser** dialog box or by typing it in the **Run Command Parameters** field. You must specify the location of any file in this field using an Express directory specifier.

- **Description**

Enter the name which you want to appear under the icon in the **Description** field. The default entry in this field will be the program filename.

You can place an additional icon on the Start Programs menu using the **Advanced** icon settings tab.

Advanced Settings Tab

The **Advanced** icon settings tab allows you to select an alternate working directory, choose an image from a separate resource, specify a shortcut key, or place an icon directly on the Start Programs menu (Windows 95 only).

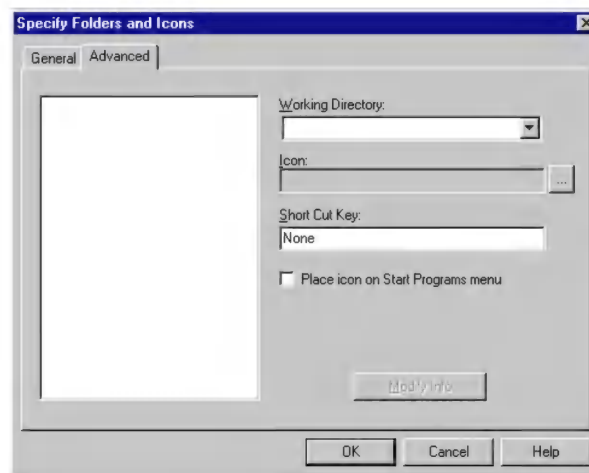


Figure 8.52. The Advanced tab of the Specify Folders and Icons dialog box.

The icons that have been added in the **General** tab are displayed in the window to the left of the edit fields. None of the advanced options are available until add an icon in the **General** tab.

Note You cannot add or delete icons from the **Advanced** icon settings tab. You must use the **General** icon settings tab.

The **Advanced** tab displays the following fields:

- **Working Directory**

The current working directory will be displayed in the **Working Directory** combo box. You can specify a different directory using Express directory specifiers. To enter the changed directory, click the **Modify Info** push button.

- **Icon**

You can select a different icon to be displayed. The resource (such as an .ICO or .DLL file) containing the icon must have already been copied to one of your program groups. Highlight the icon for which you want to set a new image. Click on the browse (...) button next to the **Icon** field to open the **Setup Files Browser** dialog box. Select the desired resource file and click **OK**.

When you return to the **Advanced** tab, the resource file will be listed in the **Icon** field. Click the **Modify Info** push button to apply the new icon.

- **Short Cut Key**

Express enables you to assign a short cut key to an icon. To specify a short cut key, Highlight the icon for which you want to set a short cut. Click on the **Short Cut Key** field so that the cursor appears in it. This is a special edit field. On your keyboard, press the key combination as you would if you were using the short cut. The **Short Cut Key** field will reflect your typing. For example, if you press the Control key, Alt key, and the letter A, the edit field will display Ctrl + Alt + A. If you make a mistake, hit the Delete key and start over.

Note If you wish to use a key on the number pad, you need to have the NumLock key on to ensure the correct key code is assigned to the short cut.

When you have entered the desired combination, click the **Modify Info** push button.

- **Place icon on Start Programs menu**

If your application only has a single icon, it will place it automatically on the Start Programs menu. If you feel the need to add additional icons to the Start Program menu, highlight the icon and select the **Place icon on Start Programs menu** check box.

Note Please note that Microsoft's Windows 95 Setup Guidelines recommend that you do this if you only have one icon, in order to make your application as easy as possible to access. However, you should **not** add more than one icon to the Start Programs menu, as that clutters up the menu.

Run Disk Builder

The Run Disk Builder section of the Setup Checklist is a single dialog box containing three tabs.

Disk Builder Tab

The **Disk Builder** tab creates a data file from your settings which will be passed to the InstallShield script, compresses your application and installation files, splits the compressed files, and places the split files into separate directories which correspond to your distribution disks.

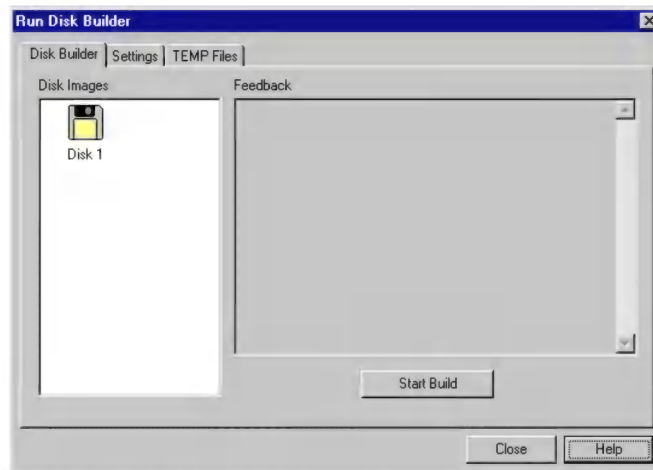


Figure 8.53. The Disk Builder tab of the Disk Builder dialog box.

Note The settings that are used in the **Disk Builder** tab are accessible in the **Settings** and **TEMP Files** tabs of this dialog box.

The **Disk Builder** tab consists of the following areas:

- **Disk Images**
This window displays an icon for each individual disk image created during the disk building process.
- **Feedback**
As the Disk Builder begins each task, it will display a message string in the Feedback window. It will also display warning and error messages as

necessary. The progress bar displayed in a secondary dialog indicates the status of the build. The builder will display a message box letting you know when it is finished.

If the Disk Builder is unable to locate a file in the directory specified, a dialog box will give you the option of launching the **Open** dialog box, which you can use to specify the location of the file. If you are unable to find the file using the dialog, you will get an error message. In this case, find the path specified for the file in the **Groups and Files** dialog box. You must then either place the file in the specified location and re-run the Disk Builder, or delete the file altogether from your setup. For an explanation of any error messages displayed during disk building, refer to Appendix C, "Disk Builder Warning Messages."

For additional information on the operation of Disk Builder, refer to Chapter 2, "A Quick Tutorial."

Settings Tab

The **Settings** tab enables you to select the size of the distribution media and set aside space on the first disk for any uncompressed files you might need to include.

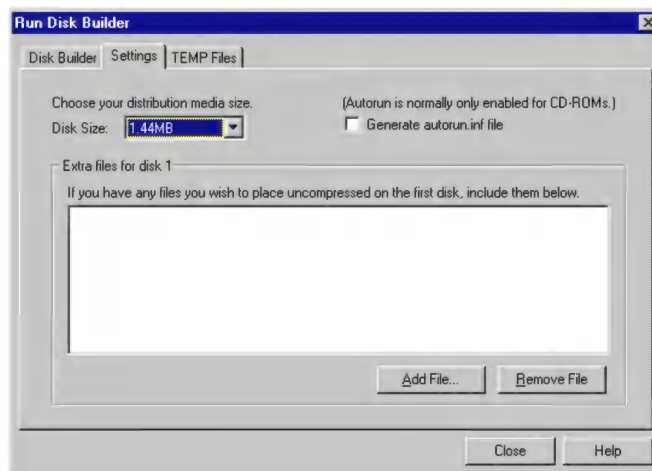


Figure 8.54. The Settings tab of the Disk Builder dialog box.

The **Settings** tab consists of the following areas:

- **Disk Size**
To select the distribution media format, choose the disk size (CD-ROM, 120 MB, 2.88MB, 1.44MB, 1.2MB, or 720K) in the **Disk Size** drop-down box.
- **Generate autorun.inf file**
If your application will be loaded from a CD-ROM drive and you would like the installation to launch automatically when the disk is placed in the drive, select the **Generate autorun.inf file** check box.
- **Extra files for disk 1**
Express provides you the option of reserving room on the first disk image for any uncompressed file you might need. This feature will be very useful if you are installing from a CD-ROM and you would like to place some of your application's files on the customer's system and leave some files on the CD-ROM or if you are using an InstallShield Extensions to access a file on your disk during the file transfer process. For more information on the use of this feature, refer to Chapter 5, "InstallShield Extensions."

TEMP Files Tab

The **TEMP Files** tab allows you to identify those files which are needed only during the installation.

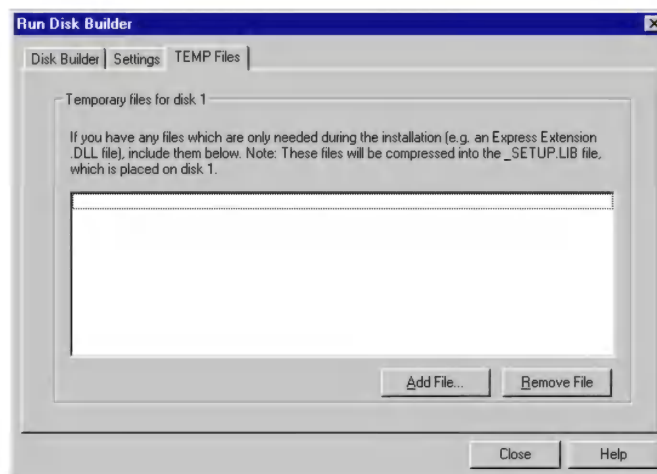


Figure 8.55. The TEMP Files tab of the Disk Builder dialog box.

The **TEMP Files** tab consists of the Temporary file for Disk 1 window. If you are using an InstallShield Extension for a .DLL file, you will want to add the .DLL file here. Express will compress any file listed in this window into the _SETUP.LIB file, which will be placed on disk 1 of your installation set. After the installation runs, the files that are included here will be deleted from your customer's system.

For more information on InstallShield Express Extensions, refer to Chapter 5, "InstallShield Extensions."

Test the Installation

The Test the Installation section of the Setup Checklist has no dialog boxes. Clicking the **Test Run** push button from the Checklist launches a test run on your machine of the installation based on the disk images you have created. For more information on testing the installation, refer to Chapter 2, "A Quick Tutorial."

Create Distribution Media

The Create the Distribution Media section is the final step in creating your installation. The disk images that you created in Disk Builder and tested are copied onto the physical distribution disks.

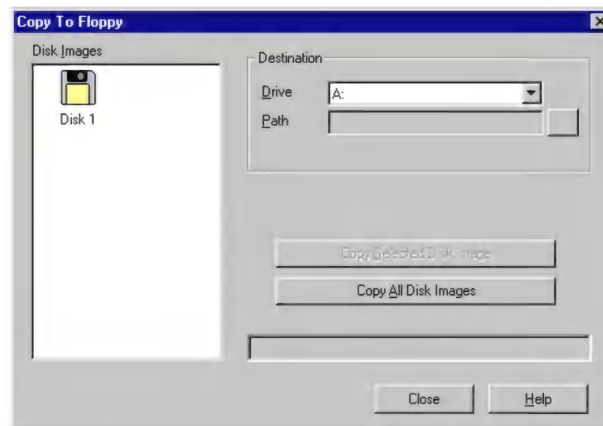


Figure 8.56. The Copy to Floppy dialog box.

The Copy to Floppy dialog box has the following components:

- **Disk Images**

The icons displayed in the Disk Images window are those that were built by Disk Builder according to your specifications.

- **Drive**

The selection in the **Drive** box determines the destination of your installation files. For example, if you select A:, Express will copy your setup directly to a disk in the A: drive.

One of the options available in the **Drive** combo box is **Path for a 1 File Installation**. Express enables you to create a single file installation for your application. A default path for the file and a password field appear on the tab when this option is selected. For more information on creating a single file installation, refer to Chapter 3, "Building Your Installation."

Note The entries in this combo-box are dynamic and reflect the disk drives on your system. For example, if your machine has two disk drives, A: and B:, both will be offered as options in the **Drive** box, along with the two path options.

- **Path**

The Path field enables to enter the path where you want to copy your setup, allowing you to place the files directly onto a network or hard disk location. You can use standard DOS paths or UNC paths in the edit field.

These options allow you to copy all or part of your setup to either disks or a network location with just a few mouse clicks. For more information on creating your distribution disks, refer to Chapter 2, "A Quick Tutorial."

Appendix A

Glossary

The following list of terms appear in the preceding text of this guide. They are provided here, in alphabetical order, for your reference.

A

alias

An alias is a set of information for a server database, usually including the database name, server name, driver type, login ID, and so on.

API (Application Programming Interface)

The Microsoft Windows API comprises some 700 functions for creating graphical user interface objects.

applet

A small, single-purpose utility application. Examples of Windows applets are CLOCK.EXE (the Windows clock) and CALC.EXE (the Windows calculator).

application file

A program or data file used in your application. When you finish writing your installation script, copy your application files and the InstallShield files to the distribution disks you ship to your customers.

Application Programming Interface

See API.

B

BDE/IDAPI (Borland Database Engine/Integrated Database Application Programming Interface)

IDAPI provides the Borland Database Engine access to multiple databases through the use of a standardized communication protocol.

billboard

A textual or graphical message displayed during the file transfer process. These billboards can be used to inform, entertain, and advertise your products to your customers.

bitmap

A binary representation of a graphical image. The term bitmap is used to refer both to the binary file (see .BMP file) and the image it creates.

.BMP file

Files with a .BMP extension contain bitmaps.

C**.CFG file**

Files with a .CFG extension contain alias settings.

check box

In a dialog box, a check box is a small, square field that the user can click “on” or “off” with the mouse. When “on,” it displays an X or check in the square, when “off,” the square is blank. Check boxes are non-exclusive—they can be turned on and off independently of one another.

combo box

In a dialog box, a combo box allows the user to either type an entry directly into the edit field or select an entry from the available drop-down list.

component

An installation component represents a group of related files. Components are the building blocks for setup types.

CONFIG.SYS

The CONFIG.SYS file contains information about the machine configuration, including device drivers, the type of keyboard, and the amount of memory set aside for disk buffers. The CONFIG.SYS file is read only when the machine boots up.

D**default directory**

The directory in which the operating system or program looks for or places files if no directory is specified.

destination directory

See target directory.

dialog box

A dialog box is a window that appears in order to exchange information with the user. When the user supplies the necessary response, the dialog box disappears.

directory specifiers

See Express directory specifiers.

distribution disk

A disk or CD-ROM that contains all or part of your compressed application files and the InstallShield Express files required to install your program on your customer's machine. The distribution media contain the files you ship to your customer.

dithered

A dithered background contains two or more colors intermingled to create either the appearance of a color not included in the palette or a gradient background.

DLL (Dynamic Link Library)

An executable file which can be accessed by any Windows-compatible application. DLLs provide an efficient means of storing various resources. DLL files typically have a .DLL extension. You can call DLLs in Express using InstallShield Express Extensions.

drag and drop

A term that refers to the Windows process of transferring files by highlighting them, then, with the left mouse button depressed, dragging them to their new location. Express allows you to add files to groups by using the drag and drop technique.

dynamic-link library

See DLL.

E**edit field**

A field in a dialog box into which the user can type a text entry.

environment variable

A string that contains a drive, path, filename, or other environment information. In Express, you can set the value of an environment variable in the AUTOEXEC. BAT dialog.

.EXE file

An .EXE (executable) file contains a relocatable machine code program. To execute the program, you need type only the name of the file, without the .EXE extension.

Express directory specifiers

Directory specifiers are used whenever you need to express the location of a file after it has been copied to the target system. InstallShield Express includes a set of directory specifiers which are replaced by information about the user's system during the installation.

F**FTP** (File Transfer Protocol)

An Internet protocol that allows a user on one host to transfer files to and from another host over a network.

G**graphical user interface**

The part of the program that interacts with the user. The graphical user interface, sometimes referred to by the acronym GUI, uses images such as windows and icons to represent program features.

GUI

See graphical user interface.

I**.ICO file**

A file containing the binary representation of an icon.

L**library file**

A file containing a collection of files.

list box

In a dialog box, a list box is an area in which the user can choose among a list of items, such as files or directories.

long filename

A filename, path, or fully qualified path and filename conforming to size and character limits exceeding those of the traditional DOS-based 8.3 filenames. Long filenames can have more than 8 characters in the name and more than 3 characters in the extension. They can use all the characters in an 8.3 format filename, as well as plus signs (+), commas (,), semicolons (;), equal signs (=), left and right square brackets ([]), and spaces. Long filenames can be up to 255 characters long.

M**main window**

The primary object in the graphical user interface. Other objects used in the installation, such as dialog boxes, are all contained within the main window.

O**object linking and embedding**

See OLE.

.OCX

Files with this extension are OLE custom controls, which are quickly replacing Visual Basic controls in many new applications.

ODBC (Open Database Connectivity)

ODBC provides access to multiple databases through the use of a single API.

OLE (Object Linking and Embedding)

A set of standards, developed by Microsoft, used to create dynamic, automatically updated links between documents, and to embed documents created by different applications.

P**progress bar**

The actual sliding bar in the progress indicator which is intended to gauge the progress of the file transfer operations.

progress indicator

A panel which you can activate during the file transfer operations to display the progress of the installation.

push button

A small area within a window which the user clicks on with the mouse button in order to make a specified selection.

R**radio button**

In a dialog box, radio buttons are small, circular fields, only one of which can be chosen at a time. The chosen button is filled with a black circle, and the others remain white. Radio buttons are exclusive—choosing any button with the mouse causes all other buttons in the set to be cleared.

registry

A central storage area which contains information about Windows applications and allows integration of applications using OLE.

registry key

An entry in the registry containing information about a specific application.

registry values

The settings assigned to a registry key.

ReportSmith

Express' ReportSmith dialog allows you to install database connections to access the ReportSmith report writer, as well as specify any necessary parameters for your connections.

resource

A modifiable part of an application program or operating system such as a menu, icon, bitmap, or text string.

RLE (run length encoding)

A method of encoding a bitmap file to reduce its storage size. Express does not support bitmaps compressed using RLE.

S**self-extracting executable file**

One of the distribution methods provided in Express. You can create a installation program that installs your application by launching an single file. This is a very popular method of distributing applications across a network or from a FTP site.

setup checklist

The Express interface tool that helps guide you through the installation creation process.

setup project file

The file created by Express which contains all of the selections and settings you select in creating your installation.

setup type

A set of components which represents all of the files to be installed. A Typical setup type usually includes all components, a Compact setup type includes only the minimum required components, and a Custom setup type allows the user to select which components will be installed.

SETUP.LIB

The library file created by Express containing the compressed resources for the installation, including temporary installation-related resources such as bitmap images and .DLLs. This file is deleted when the installation is concluded successfully.

shared .DLL

A .DLL which is in use by multiple applications. A shared .DLL will be locked if it is in use by another application while an installation is underway, making updating the .DLL from the installation impossible. This is why you should recommend that all other programs be closed before installing a new product.

source directory

The directory from which you are copying or moving files.

specifiers

See Express directory specifiers.

SQL (Structured Query Language)

A data access language used by many database server engines to retrieve data from client applications.

Start Programs menu

Windows 95's drop-drop menu of programs revealed by clicking on the Start button. Express enables you to add your program to the user's Start Program menu automatically during the installation process. This can be selected through the Advanced icon settings dialog.

startup message

The dialog box displayed by Express at the start of the installation. The startup message informs the user that the installation is initializing, and displays a progress bar during the initialization.

T**target directory**

The directory to which a file or files are copied. Target directories are signified in Express using the Express directory specifiers.

U**UNC** (Universal Naming Convention)

A standard for naming file pathnames, which allows access to network resources not mapped to local drives.

uninstallation

Uninstallation capability is one of the Windows 95 Logo Requirements for your application. To help you comply with this, Express comes complete with unInstallShield.

V**version number**

Express enables you to manually set your application's version number, or to accept the default entry taken from the version resource for your main executable file.

W**.WMF file**

The .WMF extension designates a Windows metafile. InstallShield Express supports the use of "Placeable Windows Metafiles," not "Standard Windows Metafiles."

Appendix B

InstallShield Express License Agreement

A copy of the InstallShield Installation System License Agreement follows.

IMPORTANT – READ CAREFULLY BEFORE INSTALLING THE SOFTWARE. When you enter your name as required during installation, you indicate your acceptance of the following InstallShield Express License Agreement.

This is a legal agreement between you (either an individual or an entity) and InstallShield Corporation ("INSTALLSHIELD"). Use of this software constitutes acceptance of the terms of this agreement. If you do not agree to the terms of this agreement, contact InstallShield for other licensing options or promptly return the software and the accompanying items (including written materials and binders or other containers) to INSTALLSHIELD or its authorized distributor from which you received the software for a full refund.

1. GRANT OF LICENSE

- a. The enclosed INSTALLSHIELD software program (the "Licensed Software") may be used only by the registered developer. You may use one copy of the InstallShield software product identified above, which includes "online" or electronic documents (the "SOFTWARE") on a single computer. The SOFTWARE is in "use" on a computer when it is loaded into temporary memory (i.e. RAM) or installed into permanent memory (e.g., hard disk, CD-ROM, or other storage device) of that computer.
- b. You may use the Licensed Software solely for the purpose of creating and distributing application installation systems, including updates and revisions of the single installation system.
- c. You have a royalty free right to reproduce, and distribute, in object code form and solely as part of your installation system, the installation executable files included in the Licensed Software and specifically identified in the documentation, provided you (i) include your valid copyright notice on your application; and (ii) agree to indemnify, hold harmless and defend INSTALLSHIELD from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of your application. You may not, however, modify the INSTALLSHIELD copyright notice embedded in the resources contained in the installation executable files.

2. COPYRIGHT

The Licensed Software is owned by INSTALLSHIELD and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the Licensed Software like any other copyrighted material (e.g., a book or musical recording), except that you may either (i) make one copy of the Licensed Software solely for backup or archival purposes, provided you reproduce and include INSTALLSHIELD's copyright and trademark notices contained on the original disk labels on such backup copy, or (ii) transfer the Licensed Software to a single hard disk, provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the Licensed Software.

3. OTHER RESTRICTIONS

- a. You may not use, copy, or modify the files containing the licensed software, or any backup copy, in whole or in part, or translate such files into any other file format or language, except as expressly provided for in this agreement. You may not rent, lease or sublicense the Licensed Software, but you may transfer the Licensed Software and accompanying written materials on a permanent basis, provided you retain no copies, the recipient agrees to the terms of this License Agreement, immediately contacts INSTALLSHIELD to inform INSTALLSHIELD of such transfer and pays to INSTALLSHIELD a transfer fee in an amount to be determined by INSTALLSHIELD and in effect at the time in question. Upon such transfer, your license under this License Agreement is automatically terminated.
- b. You may not reverse engineer, decompile or disassemble the Licensed Software for any purpose whatsoever.

4. TERM

Your license is effective upon payment for and the opening of the package containing the Licensed Software. You may terminate it at any time by destroying the Licensed Software together with all copies. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy all copies of the Licensed Software in any form in your possession or under your control.

5. LIMITED WARRANTY

INSTALLSHIELD warrants that for a period of ninety (90) days from the date of receipt, the Licensed Software will perform substantially in accordance with the accompanying written materials. THE FOREGOING IS THE SOLE AND EXCLUSIVE WARRANTY OFFERED BY INSTALLSHIELD. INSTALLSHIELD DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE LICENSED SOFTWARE AND ALL ACCOMPANYING MATERIALS.

6. CUSTOMER REMEDY

INSTALLSHIELD's entire liability and your exclusive remedy shall be, at INSTALLSHIELD's option, either (a) return of the price paid, or (b) repair or replacement of the Licensed Software that does not meet INSTALLSHIELD's Limited Warranty and which is returned to INSTALLSHIELD within the warranty period. The Limited Warranty is void if failure of the Licensed Software has resulted from accident, abuse or misapplication. Any replacement of Licensed Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.

7. DISCLAIMER OF DAMAGES

You assume responsibility for, among other things, (i) the selection of the Licensed Software to achieve your intended results, (ii) the acquisition of other software (including any programming or operating system software) and/or equipment compatible with the Licensed Software, and (iii) the installation, use and results obtained from the Licensed Software. Further, inasmuch as the price paid for the license rights granted to you to use the Licensed Software may be substantially disproportionate to the value of products to be used in conjunction with the Licensed Software, and for the express purpose of limiting the liability against INSTALLSHIELD to an extent which is reasonably proportionate to the commercial value of this transaction, you agree that, to the maximum extent permitted by law, INSTALLSHIELD shall in no event be liable for any damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use or inability to use the Licensed Software, whether direct, indirect, incidental, consequential, special or otherwise, REGARDLESS OF THE FORM OF ACTION, even if INSTALLSHIELD has been advised of the possibility of such damages.

8. UPDATES & TECHNICAL SUPPORT

Upon your return of a fully completed InstallShield Express Professional Product Registration Form and for a period of thirty (30) days from the date of your receipt of the Licensed Software, INSTALLSHIELD will make available to you Technical Support in the manner and under the guidelines set forth in the Licensed Software user documentation, which may be modified from time to time by INSTALLSHIELD at its discretion without notice. INSTALLSHIELD may, from time to time, revise or update the Licensed Software. In so doing, INSTALLSHIELD incurs no obligation to furnish such revision or updates to you. Updates and further support terms are available to you on the same basis as INSTALLSHIELD makes them available to its other licensees at then current prices.

9. GENERAL

- a. This License is personal between you and INSTALLSHIELD. It is not transferable and any attempt by you to rent, lease, sublicense, assign or transfer any of the rights, duties or obligations hereunder, except as provided in Section 3, above, is void. This Agreement and the conduct of the parties hereto shall be governed by the laws of the State of Illinois.
- b. YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN YOU AND INSTALLSHIELD WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN YOU AND INSTALLSHIELD OR INSTALLSHIELD'S AGENT(S) RELATING TO THE LICENSED SOFTWARE.

Appendix C

Disk Builder Warning Messages

Warning Messages

Warning messages are displayed in the Feedback window of the Disk Builder. The warning will not cause the Disk Builder to stop executing, but the condition that caused the warning may cause serious problems in your setup.

Many warning messages concern the misuse of Express directory specifiers. A warning will occur if an unknown group or variable name is referenced.

Note Group name directory specifiers are delimited by [] and the other directory specifiers by < >. The group names and specifiers are case- and white space-sensitive.

Warning Number and Description	Possible Solution
100 Unknown specifier <filename> used for filename in private INI file <filename>.	Go to the Private .INI Files tab of the Make System File Changes dialog box. Find your entry for the .INI file specified, and fix or delete the unknown specifier.
101 Unknown specifier <filename> used for value in section [.INI section name] of <filename>.	Go to the System .INI Files tab of the Make System File Changes dialog box. Find your entry for the .INI file specified, and fix or delete the unknown specifier.
102 Unknown specifier <filename> used for value in registry key <key name>.	Go to the Registry - Keys tab of the Make Registry Changes dialog box. Find and highlight the registry key specified. Click on the Registry - Values tab. Fix or delete the unknown specifier.

Warning Number and Description	Possible Solution
103 Unknown specifier <filename> used as command for icon <icon resource file>.	Go to the General tab of the Specify Folders and Icons dialog box. Highlight the icon specified. Fix or delete the unknown specifier in the Run Command field.
104 Unknown specifier <filename> used as parameter for icon <icon resource file>.	Go to the General tab of the Specify Folders and Icons dialog box. Highlight the icon specified. Fix or delete the unknown specifier in the Run Command Parameters field.
105 Unknown specifier <filename> used as working directory for icon <icon resource file>.	Go to the Advanced tab of the Specify Folders and Icons dialog box. Highlight the icon specified. Fix or delete the unknown specifier in the Working Directory field.
106 Unknown specifier <filename> used as icon file name for icon <icon resource file>.	Go to the Advanced tab of the Specify Folders and Icons dialog box. Highlight the icon specified. Fix or delete the unknown specifier in the Icon edit field.
107 Unknown specifier <filename> used as environment variable in AUTOEXEC.BAT section.	Go to the AUTOEXEC.BAT tab of the Make System File Changes dialog box. Find the unknown specifier in the Environment Variable window. Fix or delete the entry for the variable.
108 Unknown specifier <filename> used as command in AUTOEXEC.BAT section.	Go to the AUTOEXEC.BAT tab of the Make System File Changes dialog box. Find the unknown specifier in the Command field. Fix or delete the entry.

Warning Number and Description	Possible Solution
109 Unknown specifier <filename> used as setting in CONFIG.SYS section.	Go to the CONFIG.SYS tab of the Make System File Changes dialog box. Find the setting containing the unknown specifier in the <u>C</u> ONFIG.SYS Changes window. Highlight the setting, then fix or delete the entry in the Line to add field.
110 Unknown specifier <filename> used as run command in Setup Complete dialog box.	Go to the Setup Complete user dialog box in the Select User Interface Components section and click on the Settings tab. Fix or delete the unknown specifier in the Run Command field.
111 Unknown specifier <filename> used as run command parameter in Setup Complete dialog box.	Go to the Setup Complete user dialog box in the Select User Interface Components section and click on the Settings tab. Fix or delete the unknown specifier in the Run Command Parameters field.
112 Unknown specifier <filename> used as Readme file in Setup Complete dialog box.	Go to the Setup Complete user dialog box in the Select User Interface Components section and click on the Settings tab. Fix or delete the unknown specifier in the Readme File field.
113 Unknown specifier <filename> used as application EXE name in App Info section.	Go to the App Info tab of the Set the Visual Design dialog box. Fix or delete the unknown specifier in the Application Executable field.
114 Unknown specifier <filename> used as installation's destination directory.	Go to the Groups tab of the Specify Components and Files dialog box. Fix or delete the unknown specifier in the Destination Directory field.

Warning Number and Description	Possible Solution
115 Missing information in Application Name, Version, or Company field of the App Info dialog box.	Go to the App Info tab of the Set the Visual Design dialog box. Add the missing information in the <u>A</u>pplication Name , <u>V</u>ersion , or <u>C</u>ompany field.
116 Unknown specifier <filename> used as filename for InstallShield Extension.	Go to the Settings tab of the Express Extension dialog box. Fix or delete the unknown specifier in the <u>F</u>ilename field.
117 Unknown specifier <filename> used as parameter for InstallShield Extension.	Go to the Settings tab of the Express Extension dialog box. Fix or delete the unknown specifier in the <u>O</u>ptional Program <u>P</u>arameters field.
118 The filename in the Express Extension <extension name> is being referenced before the file is copied. This will cause the extension to fail.	Go to the TEMP Files tab of the Disk Builder dialog box. Add the filename to the Temporary files for disk 1 window. -OR- Go to the Ordering tab of the Express Extension dialog box and move the point where the extension is invoked to later in the installation.
120 The group [group name] is not assigned to any component.	Go to the Components tab of the Specify Components and Files dialog box and assign the specified group to the appropriate component.
121 The component [component name] is not assigned to any setup type.	Go to the Setup Types tab of the Specify Components and Files dialog box and add the specified component to the appropriate setup type(s).

Warning Number and Description	Possible Solution
130 No bitmap specified for Welcome Bitmap dialog box.	Go to the Welcome Bitmap Settings tab of the Dialog Boxes dialog box and specify the appropriate bitmap file.
131 No file specified for Software License Agreement dialog box.	Go to the Software License Agreement Settings tab of the Dialog Boxes dialog box and specify the file containing your license agreement.
132 No file specified for Readme Information dialog box.	Go to the Readme Information Settings tab of the Dialog Boxes dialog box and specify your Readme file.
133 No default directory specified for Choose Destination Location dialog box.	Go to the Choose Destination Location Settings tab of the Dialog Boxes dialog box and specify a default directory.
134 No default folder name given for Select Program Folder dialog box.	Go to the Select Program Folder Settings tab of the Dialog Boxes dialog box and specify a default folder name.
135 No directory given for billboard file(s).	Go to the Billboards Settings tab of the Dialog Boxes dialog box and specify the directory which contains your billboard file(s).
136 The file <filename> has a long filename which is not supported in Win 3.x.	You must remove or rename the specified file so that its fully qualified path does not contain any long filenames (16-bit setups only).
140 No icons created as part of your setup.	Go to the General tab of the Specify Folders and Icons dialog box. Specify the icon(s) you want to include in your program folder.

Warning Number and Description	Possible Solution
141 The icon description <icon filename> is too long for Program Manager.* Program Manager limits icon descriptions to 40 characters.	Go to the General tab of the Specify Folders and Icons dialog box. Edit your entry in the Description field until it contains 40 or fewer characters.
143 The <INSTALLDIR> directory specifier should not be used in the Choose Destination Location user dialog box.	Go to the Choose Destination Location Settings tab of the Dialog Boxes dialog box and specify a default target directory that does not include the <INSTALLDIR> directory specifier.
144 No components were added to the [group name] setup type.	Go to the Setup Types tab of the Specify Components and Files dialog box and add the appropriate component(s) to the specified setup type.
145 No groups were added to the [component name] component.	Go to the Components tab of the Specify Components and Files dialog box and either add the appropriate groups to the specified component, or delete the component.
146 No files were added to the [group name] group.	Go to the Groups and Files tab of the Specify Components and Files dialog box and either add the appropriate files to the specified group, or delete the group.

* Program Manager applies not only to Windows 3.1, but also to Windows NT and, in some cases, Windows 95.

Error Messages

An error message from the Disk Builder will cause Express to stop executing. Error messages are generated by problems which prevent one of the InstallShield Express programs from carrying out its task. Except for error number 150, errors are displayed in a secondary dialog box, as shown below:

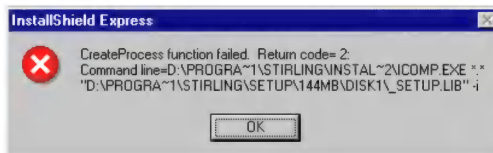


Figure C.1. A typical error message generated by the Disk Builder.

A list of error messages generated by InstallShield Express appears below. Where available, a solution to the error is provided.

Error Number	Description
-1	Unknown error
-2	Unable to open input file. Command <filename>
-3	Unable to open output file. Command <filename>
-4	Unable to write to file. Command <filename>
-5	Input file not compressed with InstallShield Compressor
-6	Memory allocation error
-7	Header information of compressed file is incorrect
-9	Source and target directories conflict
-16	Library compacting error
-38	Out of disk space

Solution You must free disk space on the drive containing your setup project before running the Disk Builder again.

Error Number	Description
-43	Compression error during read/write of non-compression data
-46	Target file is read-only

Solution Check that a project file does not already exist with the same name in a read-only format.

-150 Build canceled due to missing files.

Solution When Express is unable to locate a file in the path specified, a dialog box will give you the option of launching the Open dialog box, which you can use to locate the file. If you are unable to find the file using the dialog box, you will get this error message. You must then either place the file in the location specified in the **Groups and Files** dialog box or delete it from your setup.

Appendix D

Index

.

.BMP, A-2
 .CFG, A-2
 .CFG file, 4-24
 .DLL, 2-11
 Calling a Custom, 5-4
 definition, A-3
 self-registration, 7-9
 shared .DLL, 3-24, A-7
 .EXE
 definition, A-3
 Running an, 5-5
 .ICO, A-4
 .INI, 6-2
 .IWZ, 8-2
 .OCX, 8-14, A-5
 self-registration, 7-9
 .VBX, 8-14
 .VXD, 8-40
 .WMF, A-8

—

_SETUP.LIB, 3-22, 5-2, 8-25, 8-27, 8-28,
 8-61, A-6

<

<CommonFilesDir>, 3-22, 8-31
 <INSTALLDIR>, 3-15, 3-21, 5-5, 8-31
 <ProgramFilesDir>, 3-22, 8-31
 <SRCDIR>, 3-22, 5-4
 <SUPPORTDIR>, 3-22, 5-4
 <WINDIR>, 3-21
 <WINDISK>, 3-21
 <WINSYS16DIR>, 3-21
 <WINSYSDIR>, 3-15, 3-21, 3-29
 <WINSYSDISK>, 3-21

1

16-bit, 3-5

3

32-bit, 3-5
 3D Controls, 4-18

A

Advanced Settings tab, 8-55
 alias, A-1
 Always show reboot computer dialog, 8-40
 Animated Button Control, 4-17
 API, A-1
 App Info tab, 6-8, 8-4, 8-35
 applet, A-1
 Application Components window, 8-21,
 8-22
 Application Executable field, 8-5
 application file, A-1
 Application Information, 2-6, 8-30
 Application Information key, 6-8
 Application Name field, 8-4
 Application Programming Interface. See
 API
 Application Uninstallation key, 6-9
 AUTOEXEC.BAT
 file, 2-29, 3-24, 8-47
 tab, 8-47
 Automatic Uninstaller, 8-8

Contents				A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q	
R	S	T	U	V	W	Special Characters				

B

Background color field, 8-7
 baggage file, 8-25, 8-38
 base keys, 6-4, 6-6
 BDE, 1-8, 4-5, 4-8, 4-13
 definition, A-1
 optional alias parameters, 7-2
 BDE Alias, 7-2
 BDE Installation Type, 4-22, 4-23
 BDE/IDAPI settings, 4-22
 billboard, 2-16, 8-37, A-1
 Binary value data, 7-6
 bitmap
 definition, A-1
 using as a logo, 2-11
 Borland C++ Objects, 4-4, 8-9
 Borland custom controls. See Borland C++ objects
 Borland Database Engine. See BDE
 Borland Delphi Objects, 4-7

C

chapter descriptions, 1-12
 check box, A-2
 Choose Destination Location, 2-16, 8-4, 8-30
 ClassLib
 dynamically linked. See Borland C++ objects
 dynamically linked, multithreaded. See Borland C++ objects
 Client/Server Paradox Runtime, 4-13, 8-13
 combo box, A-2
 Comm Control, 4-18
 Command field, 8-48
 Common Dialog Control, 4-17
 Company field, 8-6
 component
 definition, A-2

Components

Name field, 8-21
 tab, 8-20
 usage in Express, 3-13
 usage in Express, 3-9
 CONFIG.SYS
 Changes window, 8-49
 file, 2-29, 3-24, A-2
 tab, 8-49
 Copy to Floppy dialog box, 2-36, 3-27, 8-62
 copyright information, ii
 Create Distribution Media, 8-61
 Crystal Report Control, 4-17
 custom registry keys
 creating, 6-10
 removing, 6-14
 Custom setup type, 2-16, 8-3, 8-33

D

DAO, 4-17, 4-19
 Data Access Objects, 1-8
 Data Bound Grid Control, 4-17
 Data Bound List Control, 4-17
 default directory, A-2
 default setting, 3-26
 Delphi Objects, 8-10
 DemoShield, 3-3
 Description field, 8-21, 8-55
 destination directory, 7-3. See target directory
 Destination Directory field, 8-19
 Dialog Box
 Create Distribution Media, 8-61
 definition, A-2
 Make Registry Changes, 8-50
 Make System Files Changes, 8-45
 New Project, 8-2
 Run Disk Builder, 8-58
 Select InstallShield Objects, 8-9
 Select User Interface Components, 7-3, 8-23

Contents	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W
Special Characters							

Set the Visual Design, 8-4
Specify Components and Files, 7-3, 8-18
Specify Folders and Icons, 8-54
directory specifiers, 3-20, A-2
Directory to Add field, 8-47
Directory window, 8-3
Disk Builder, 8-58
 tab, 8-58
disk building, 2-3
disk building incomplete, 7-4
disk images, 7-4
Disk Images window, 8-58, 8-62
Disk Size field, 8-60
Display full pathnames for files, 8-20
distribution disk, A-3
dithered, A-3
document conventions, 1-12
drag and drop, A-3
Drive field, 8-62
Drive window, 8-3
dynamic-link library. See .DLL

E

edit field, A-3
Email address. See Technical Support
environment variable
 definition, A-3
 field, 8-48
Express directory specifiers, A-4
Express Extensions, 8-42
Extensions window, 8-43
Extra files for disk 1 window, 8-60

F

Features tab, 8-7
Feedback window, 8-58
File Groups
 usage in Express, 3-9
 window, 8-18, 8-21
FTP, 3-27, A-6

Full Key Name field, 8-51

G

Gauge Control, 4-17
General Settings tab, 8-54
Generate autorun.inf file, 8-60
Graph Control, 4-18
graphical user interface, A-4
Grid Control, 4-17, 4-19
Group and Files, 7-11
group name
 field, 8-19
 usage, 3-22
Groups tab, 8-18
GUI. See graphical user interface

H

hex value data, 6-13, 7-6
HKEY_CLASSES_ROOT, 6-4, 6-6
HKEY_CURRENT_CONFIG, 6-6
HKEY_CURRENT_USER, 6-6
HKEY_DYN_DATA, 6-6
HKEY_LOCAL_MACHINE, 6-6
HKEY_USERS, 6-6

I

Icon field, 8-56
Icon Selection, 7-6
IDAPI, A-1
INI File field, 8-45
installation is not waiting for
 extension, 7-8
installation, purpose of, 3-8
InstallShield Express
 custom editions, 1-7
 features, 1-8
Functional Benefits, 3-8
Help System, 1-14

Contents				A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q	
R	S	T	U	V	W	Special Characters				

Installation, 1-11
 License Agreement, 1-16
 Online Registration, 1-11
 Product Registration, vii, 1-16
 Quick Tour, 3-2
 Readme File, 1-15
 System Requirements, 1-10
 Technical Support Request Form, 1-16
 User Interface, 1-17
 InstallShield Objects, 1-8, 4-2
 InstallShield SDK Edition, 1-2
 InstallShield3, 1-4
 Internet website. See Technical Support

J

Jet, 4-17, 4-19

K

Key State Control, 4-17, 4-19
 Keyword field, 8-45, 8-46

L

Launch application dialog window, 8-41
 library file, A-4
 Line to add field, 8-49
 list box, A-4
 Logo Bitmap window, 8-7
 long filename, A-4

M

Main Title
 window, 8-7
 Main Window
 definition, A-4
 tab, 8-6
 Make Registry Changes, 8-50
 Make System Files Changes, 8-45

MAPI Control, 4-18
 Masked Edit Control, 4-18
 Menu Bar, 1-18
 metafile, A-8
 MFC, 4-19
 Microsoft Comm Control, 4-19
 Microsoft Masked Edit Control, 4-20
 Microsoft Multimedia Control, 4-20
 Multimedia Control, 4-17
 multiple billboards, 8-38

N

Name=Value field, 8-48
 new project, 2-3, 2-4, 3-4, 8-2
 New Subdirectory field, 8-3

O

object linking and embedding. See OLE
 OCF
 linked and embedded. See Borland C++
 objects
 ODBC, 4-6, 4-8, 4-13, 4-17, 4-19, A-5
 ODBC drivers missing, 7-6
 OLE, A-5
 Online help, 1-14
 Online registration, 2-17, 8-38
 Opening a existing project, 2-11
 Ordering window, 8-43
 Outline Control, 4-18
 OWL
 dynamically linked. See Borland C++
 objects
 dynamically linked, multithreaded. See
 Borland C++ objects

P

Paradox Objects, 4-12, 8-12
 Paradox Runtime, 4-13, 8-13

Contents	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W
Special Characters							

Paradox Runtime Client/Server. See
Client/Server Paradox Runtime
Partial BDE installation, 7-2
Path field, 8-62
Path for a 1 File Installation. See Self-
extracting executable file
Per Application Paths (App Paths)
Information key, 6-8
Picture Clip Control, 4-18
Place icon on Start Programs menu, 8-57
Private.INI File
adding information, 3-25
tab, 8-45
Private INI File
Changes window, 8-46
Product information, x
progress bar, 8-37, A-5
Progress Indicator, 2-16, 8-36, A-5
Project Language field, 8-3
Project Name field, 8-2
Project Type field, 8-2
PROTOCOL.INI, 8-46
push button, A-5

Q

Quick Tour, 3-2

R

radio button, A-5
Readme file
adding a, 2-11
Express documentation, 1-15
information, 2-15
Readme Information, 8-27
REG Files tab, 8-52
registry, 3-24, 3-25, A-5
administrative tools, 6-5
application configurations, 6-4
Application Information key, 6-8
Application Uninstallation key, 6-9

base keys, 6-4, 6-6
custom keys and values, 6-10
definition, 6-2
device drivers, 6-4
function, 6-2
hardware configurations, 6-4
HKEY_CLASSES_ROOT, 6-4, 6-6
HKEY_CURRENT_CONFIG, 6-6
HKEY_CURRENT_USER, 6-6
HKEY_DYN_DATA, 6-6
HKEY_LOCAL_MACHINE, 6-6
HKEY_USERS, 6-6
keys and values, 6-5
Per Application Paths (App Paths)
Information key, 6-8
Registry - Keys tab, 8-50
Registry - Values tab, 8-51
Registry Editor, 6-5, 6-15
Registry Key field, 8-52
Registry Keys window, 2-31
subkey, 6-3
Registry Editor, 6-5, 6-15
registry key, A-5
Registry Keys window, 8-50
registry values, A-6
Remote Client Support, 4-17
Remote Data Control, 4-18
ReportSmith
definition, A-6
Runtime Report Viewer, 4-8
Settings, 2-8
resource, A-6
Rich Textbox Control, 4-18
RLE, A-6
RTL
dynamically linked. See Borland C++
objects
dynamically linked, multithreaded. See
Borland C++ objects
Run Command field, 8-54
Run Command Parameters field, 8-55
run length encoding, A-6

Contents				A	B	C	D	E	F	G
H	I	J	K	L	M	N	O	P	Q	
R	S	T	U	V	W	Special Characters				

S

Saving space on disk 1, 5-7
 Section field, 8-45, 8-46
 Select InstallShield Objects, 8-9
 Select Program Folder, 4-16, 8-34
 Select User Interface Components, 8-23, 8-43
 self-extracting executable file, 2-37, 3-27, 7-3, A-6
 self-registration of .DLLs and .OCXs, 7-9
 Set the Visual Design, 8-4
 Settings tab, 8-59
 Setup Checklist, 1-17, A-6
 Setup Complete, 2-17, 8-40
 Setup Components, 3-13
 Setup Groups, 3-15
 setup initialization error, 7-7
 setup project file
 closing, 3-7
 definition, A-6
 opening existing, 3-7
 opening new, 3-2
 saving, 3-7
 Setup Type, 2-16, 2-26, 8-32, A-6
 setup types, 3-10
 compact, 3-11
 custom, 3-11
 single, 3-13
 typical, 3-11
 usage in Express, 3-9
 Setup Types tab, 8-21
 Setup Types window, 8-22
 SETUP.EXE, 3-26
 Short Cut Key field, 8-56
 single setup. See setup types
 Software License Agreement, 2-15, 8-26
 Source directory, 3-20, A-7
 specifiers. See directory specifiers

Specify Components and Files, 8-18
 Specify Folders and Icons, 8-54
 Spin Button Control, 4-18
 SQL
 definition, A-7
 SQL Links, 4-5, 4-8, 4-13, 4-27
 SQL Server, 7-2
 Start Copying Files, 2-16, 8-36
 Start Programs menu
 definition, A-7
 Start Programs menu, adding icons, 3-25
 startup message, A-7
 subdirectories, 7-10
 subkey, 6-3
 Symbols
 Bent corner, 1-20
 Checkmarks, 1-20
 Push Button, 1-20
 Status bar, 1-20
 System .INI Files tab, 8-46
 System File Changes window, 8-47
 System menu, 1-18

T

Tabbed Dialog Control, 4-18
 Target directory, 3-19, A-7
 target platform
 16-bit or 32-bit, 3-5
 field, 8-3
 window, 8-8
 Technical Support
 Contact information, 1-x
 Request form, 1-16
 Support via Internet, 1-x
 TEMP Files tab, 8-60
 Temporary files for Disk 1 window, 8-61
 Test the installation, 2-3, 8-61
 Title Bar, 1-18
 Toolbar, 1-18
 About button, 1-19
 Build button, 1-19

Contents	A	B	C	D	E	F	G
H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W
Special Characters							

Copy to Floppy button, 1-19
 Explorer button, 1-19
 Help button, 1-19
 New button, 1-19
 Open button, 1-19
 Save button, 1-19
 Test Run button, 1-19

version control, 3-29
 Version field, 8-5
 version number, 2-22, A-8
 Visual Basic Objects, 4-15, 8-13
 Begin Automatic Review, 8-14
 Visual Basic project file, 8-14
 Visual C++ Objects, 4-19, 8-15
 VSHARE.386, 8-48

U

UNC, 8-62, A-7
 uninstall icon, 7-12
 uninstall key, 8-5, 8-51
 uninstallation, 3-26, 7-12, A-7
 unInstallShield, 3-26, 6-9, 6-14, 8-8
 User Dialog Boxes, 8-23
 Billboards, 8-37
 Choose Destination Location, 8-30
 Custom Setup, 8-33
 Online Registration, 8-38
 Progress Indicator, 8-36
 Readme Information, 8-27
 Select Program Folder, 8-34
 Setup Complete, 8-40
 Setup Type, 8-32
 Software License Agreement, 8-26
 Start Copying Files, 8-36
 User Information, 8-28
 Welcome Bitmap, 8-23
 Welcome Message, 8-25
 User Information, 2-15, 8-28
 USER.EXE, 8-30

W

Welcome Bitmap, 2-15, 8-23
 Welcome Message, 2-15, 8-4, 8-25
 WIN.INI, 3-24, 6-2, 8-30, 8-46
 WIN.INI, 2-28, 3-25
 window background, 2-11
 window text, 2-11
 Windows 95 Setup Guidelines, 3-8, 3-24, 8-31, 8-57
 Windows Common Controls, 4-18
 Working Directory field, 8-56

V

Value field, 8-46, 8-47
 Values window, 8-52
 VBX, 4-6
 VDBT
 programmatic access, 4-6
 with OLE automation, 4-6
 with VBX, 4-6